LOCKED AT THE ENTRANCE, OPEN INSIDE: WHERE RLVR NARROWS THE SOLUTION SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) substantially improves single-sample accuracy (pass@1) but causes the policy's solution space to contract, diminishing the returns of test-time scaling. In this work, we investigate where inside a reasoning trajectory this breadth is lost: does the policy fail to **access** a valid solution family, or does it fail to **execute** computation once initiated? To disentangle *access* from *execution*, we analyze the Countdown task, whose solution space can be exhaustively enumerated into discrete **entrance families** defined by the first operand and operator, across PPO on Qwen2.5-3B and GRPO on Qwen2.5-3B-Instruct. Across both training setups, solution coverage falls by up to 67%, halving even on problems solved at both endpoints. We show that this contraction is heavily concentrated at the entrance: per-token likelihood shifts are $11\times$–$16\times$ larger prior to the first arithmetic operation than during downstream reasoning. Supplying only an unselected entrance prefix restores completion rates in low-access families by over an order of magnitude ($0.018 \rightarrow 0.212$ under PPO), demonstrating that alternative solutions remain executable but are no longer initiated. Guided by this localization, layer interpolation restores entrance diversity and increases solution coverage by 37%, with pass@1 of 0.305 versus 0.278 for the late policy. Finally, we show that early-step entropy collapse recurs across six math benchmarks with 7B and 14B models, but is not an inevitable byproduct of reasoning optimization: an SFT baseline preserves more than double the coverage, and staged SFT–DPO–RLVR pipelines retain early-step entropy. In summary, reasoning breadth is lost at the door, not inside the room.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become a standard paradigm for eliciting complex mathematical reasoning in LLMs (Shao et al., 2024; Guo et al., 2025; Yu et al., 2025; Zeng et al., 2025; Yan et al., 2026). In parallel, inference-time compute techniques, such as repeated sampling, self-consistency, and verifier-guided tree search, rely on generating diverse candidate trajectories to boost task accuracy (Brown et al., 2024; Wang et al., 2023; Huang et al., 2025; Snell et al., 2025). However, these two directions exist in fundamental tension: while RLVR substantially increases single-sample accuracy, recent studies show that post-RLVR policies collapse into narrow answer supports and exhibit severely degraded reasoning diversity (Yue et al., 2025; Wu et al., 2026; Matsutani et al., 2026; Saha et al., 2026), which sharply curtails the marginal gains of repeated sampling.

Existing efforts attempt to quantify this contraction or mitigate it via reward shaping, exploration bonuses, and altered rollout objectives (He et al., 2025; Gai et al., 2025; Li et al., 2026a; Song et al., 2025). However, where valid solutions are lost along a reasoning trajectory remains unidentified. Because prior works evaluate diversity via aggregate final-answer counts or sampled trace clustering, they conflate two distinct failure modes: a policy may fail to **access** a valid solution family (i.e., never initiating the path), or it may fail to **execute** it once accessed (i.e., derailing during downstream computation). Disentangling *access* from *execution* is essential, as the two failure modes dictate fundamentally different recovery mechanisms.

To decouple *access* from *execution*, we study the Countdown task (Pan et al., 2025), where the entire valid solution space can be exhaustively enumerated and partitioned into discrete **entrance fami-**

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

lies defined by the initial operand and operator (Figure 1). We measure free-generation coverage to quantify family *access*, and probe conditional capability by clamping the prompt to a solver-specified entrance while leaving all downstream arithmetic open to measure *execution*. We evaluate this framework across two distinct RLVR implementations: a PPO training trajectory on Qwen2.5-3B and an open-source GRPO checkpoint series on Qwen2.5-3B-Instruct.

RLVR systematically induces an accuracy–breadth tradeoff across training algorithms and checkpoints (§4). By tracking single-sample accuracy pass@1 alongside exhaustive solution-space coverage across the full test distribution throughout RLVR training, we observe an inverse scaling relationship: under PPO, pass@1 increases approximately 5.6-fold while overall solution coverage plummets from 0.337 to 0.111. GRPO checkpoint series exhibits the same contraction, with accuracy tripling while solution coverage drops by 43%. Crucially, this contraction persists on a subset of problems solved at both endpoints, demonstrating that solution breadth collapses even when underlying problem difficulty is well within the model's capability.

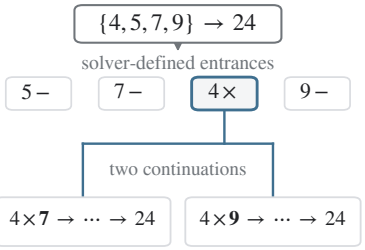

Figure 1: **Entrances partition solution set.** For $\{4, 5, 7, 9\} \rightarrow 24$, solver-defined entrances group valid solutions by their first operand and operator. Highlighted entrance $4\times$ admits distinct continuations beginning with 4×7 and 4×9.

Reasoning breadth is lost at the entrance rather than through downstream execution collapse (§5). By combining teacher-forced log-likelihood phase attribution with entrance-clamped rollouts, we isolate where probability mass shifts during training. Per-token log-likelihood divergence is $16\times$ (PPO) and $11\times$ (GRPO) larger prior to the first arithmetic operation than across all subsequent reasoning steps. Furthermore, supplying the model with a minimal, unselected entrance prefix restores completion rates in low-access families by over an order of magnitude ($0.018 \rightarrow 0.212$ under PPO), while downstream execution capability on matched prefixes strictly improves during training. Therefore, the trained policy remains capable of executing diverse solutions, and it simply stops entering them.

Targeting opening decisions recovers solution diversity (§6). Guided by our access-localization finding, we evaluate surface-level prompts and decoding shifts against interventions that redistribute early computational states. Method-prompting and reasoning-phase logit mixing yield modest coverage gains. Layer-specific parameter interpolation (blending late-checkpoint layers 20–28 with step-50 weights) increases solution coverage by 37%, with pass@1 of 0.305 versus 0.278 for the late policy. Entrance-family coverage and generated entrance entropy rise together, and previously unobserved families again produce successful rollouts.

Early entrance narrowing generalizes to multi-step math benchmarks and larger model scales (§7). By computing first-calculation entropy, same-trace likelihood profiles, and cross-checkpoint trace diversity across six standard math benchmarks (GSM8K, MATH500, Minerva Math, OlympiadBench, AMC23 and AIME24) on 7B and 14B Qwen architectures, we confirm that RLVR consistently triggers an early-stage distributional collapse. Supplying completed first calculations also recovers alternative routes on multi-step mathematics. Finally, evaluating alternative training paradigms reveals that breadth collapse is not an unavoidable byproduct of optimization: an SFT model maintains more than double the solution coverage, and the OLMo-3 SFT–DPO–RLVR ladder elevates pass@1 while preserving early-step calculation entropy.

In summary, our contributions are:

1. **Access vs. Execution Decomposition:** We introduce an exhaustive state-space decomposition framework on Countdown to separate solution initiation from downstream reasoning execution.
2. **Mechanistic Localization of Breadth Loss:** We demonstrate across PPO and GRPO that RLVR-driven diversity collapse is concentrated at the entrance ($11\times$–$16\times$ larger likelihood shift) rather than caused by downstream arithmetic failure.
3. **Inference- and Weight-Space Recovery:** We demonstrate that late-layer parameter interpolation restores entrance diversity and increases solution coverage by 37%, with pass@1 of 0.305 versus 0.278.

4. **Benchmarking & Scaling Generalization:** We confirm the entrance-narrowing phenomenon across 7B and 14B models on six standard math benchmarks, while identifying training regimes (such as staged SFT/DPO alignment) that circumvent this tradeoff.

2 Related Work

Distribution collapse in RLVR. While RLVR scales inference-time reasoning (Guo et al., 2025; Yu et al., 2025), on-policy optimization frequently triggers winner-take-all distribution collapse and degraded solution coverage (Wu et al., 2026; Yue et al., 2025; Zhou, 2026). Existing mitigations introduce smoothed rewards or exploration penalties (Gai et al., 2025; He et al., 2025; Song et al., 2025). Crucially, while prior studies assess reachability only at the prompt or final-answer level, we isolate where inside the reasoning trajectory paths are lost, disentangling early path selection (*access*) from downstream computation (*execution*).

Bifurcation points and prefix exploration. Autoregressive trajectories are disproportionately dictated by sparse, high-entropy forking tokens (Bigelow et al., 2025; Wang et al., 2025). Prior work diversifies first-token choices (Kim & No, 2026), clusters empirical rollouts (Saha et al., 2026), or guides reasoning via prefix prompts (Macar et al., 2026; Zhang et al., 2025); however, rollout-based clustering cannot detect valid branches the policy never visits. By grounding branch analysis in exhaustively enumerated solution graphs, we evaluate completely extinct solution families and isolate downstream execution capability using uninformative minimal prefixes (Appendix D.7).

Test-time scaling and weight-space recovery. Inference search and verifier scaling are fundamentally bounded by the policy's reachability support (Dragoi et al., 2025; Yu, 2025). To counteract post-training diversity loss, recent work ensembles temporal checkpoints or interpolates model weights (Dang et al., 2025; Li et al., 2026c). We provide a mechanistic explanation for these techniques: parameter interpolation restores both entrance diversity and successful access to previously unobserved branches (Appendix K extends this discussion).

3 Setup: Entrances, Access, and Execution

Task formulation and training regimes. A Countdown instance (Pan et al., 2025) requires reaching a target integer from 3–4 given numbers using basic arithmetic ($\{+, -, \times, \div\}$) and parentheses. Because solution space is exhaustively enumerable, we can trace exactly how and where probability mass disappears. We evaluate two independent RLVR pipelines: **Self-Trained PPO:** Following TinyZero (Pan et al., 2025), we train Qwen2.5-3B (Qwen et al., 2025) under PPO, evaluating the base model alongside 11 actor checkpoints from step 25 to 275. **Public GRPO Checkpoints:** We evaluate open-source `Qwen-2.5-3B-R1-countdown` series (Qwen2.5-3B-Instruct via TRL GRPO). Filtering the public 50,000-example training superset against sorted input-target semantic keys yields 135 strictly held-out, solver-feasible problems. An independent 300-problem sweep identifies step 25 as the earliest checkpoint exceeding 90% format compliance, with step 450 as the late endpoint.

Solution space and coverage. For any feasible problem x, a symbolic solver enumerates the leaf set $\mathcal{S}(x)$ of canonical valid solutions (modulo commutativity and associativity). Given n independent rollouts $Y_{1:n}$, per-instance coverage and mean test coverage are:

$$\mathrm{Cov}(x, Y_{1:n}) = \frac{\left|\mathcal{S}(x) \cap \mathrm{canon}(Y_{1:n})\right|}{|\mathcal{S}(x)|}, \qquad \overline{\mathrm{Cov}}(X) = \frac{1}{|X|} \sum_{x \in X} \mathrm{Cov}(x, Y_{1:n}), \tag{1}$$

where $\mathrm{canon}(Y_{1:n})$ maps outputs to their canonical forms. An independent enumerator verified feasibility and leaf completeness on 500 held-out instances with zero discrepancy (Appendix C.10).

Entrance families: Decoupling access and execution. We partition $\mathcal{S}(x)$ into **entrance families** $b = (o_1, \odot_1)$ defined by the initial operand and operator (e.g., $\{4, 5, 7, 9\} \to 24$ admits $5-$, $7-$, $4\times$, and $9-$; Figure 1), averaging 2.53 families per instance across 150 test problems. Canonical representatives define the coverage partition; generated entrances define rollout access (Appendix B). Fixing only the opening operation allows solving via b to factorize into an *access* phase and an *execution* phase:

$$\pi_\theta(\text{solve via } b \mid x) = \underbrace{\pi_\theta(B = b \mid x)}_{\text{access } A_t(b|x)} \cdot \underbrace{\pi_\theta(\text{valid completion in } b \mid x, B = b)}_{\text{observational execution}}. \tag{2}$$

Writing $\pi_t = \pi_{\theta_t}$, we estimate family access $A_t(b \mid x)$ via unconstrained sampling and teacher-forced prefix log-probabilities. To isolate execution independent of selection biases, we measure interventional execution capability $E_t^{\mathrm{do}}(b \mid x) = \pi_t(\text{valid completion in family } b \mid x \oplus e_b)$, where e_b is a solver-constructed minimal prefix (e.g., "`Let me try: 4 *`") terminating immediately after the operator. For a fixed problem and family, e_b is identical across solution leaves (Appendix D.1). We report both *designated-family completion* (success strictly within b) and *any-valid success*.

On-policy gradient dynamics on entrances. Because policy gradients update parameters strictly along sampled rollouts, initial sampling asymmetries compound: under-visited entrance families receive diminishing learning signals, triggering a self-reinforcing contraction of opening support even while latent downstream execution remains intact (see first-order surrogate in Appendix C.2) (Yuan et al., 2026; Zhou, 2026).

Structural proxies for standard math benchmarks. On non-enumerable math benchmarks, we approximate solution breadth via canonicalized reasoning traces. We define the *first calculation* as the earliest parsed token span evaluating an instantiated arithmetic relation; its categorical entropy serves as our scale-invariant metric for early search breadth. To isolate early versus late shifts, we evaluate identical reference traces across model pairs, bifurcating per-token log-likelihoods at the boundary of the first complete calculation.

Evaluation protocol. Countdown evaluations use $T = 0.7$, top-$p = 0.9$, and 320 rollouts across 150 (PPO, 256-token cap) and 135 (GRPO, 1,024-token cap) held-out feasible instances. Scaling and open-domain math evaluations evaluate Qwen2.5-7B/14B Base and SimpleRL (Zeng et al., 2025), DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025), and OLMo-3 7B progression (SFT, DPO, RLVR) (Team Olmo, 2025) across six benchmarks: GSM8K (500), MATH500 (500), Minerva (272), OlympiadBench (500), AMC23 (40), and AIME24 (30) (64 rollouts/instance, $T = 0.6$, top-$p = 0.95$, 16,000-token cap). Accuracy is reported via unbiased pass@k (Chen et al., 2021), entropy in nats, and confidence intervals via 1,000–10,000 bootstrap resamples (See Appendix B for details).

4 The Accuracy–Breadth Tradeoff Across RLVR Training

Inverse scaling between accuracy and solution coverage. Across both PPO and GRPO optimization runs, we observe solution-space contraction after the initial expansion phase (Table 1). In our self-trained PPO pipeline on Qwen2.5-3B, pass@1 increases approximately 5.6-fold between steps 50 and 275, while cumulative solution coverage $\overline{\mathrm{Cov}}(X)$ at $n = 320$ samples collapses by 67%, falling from 0.337 to 0.111.[1] The public GRPO series demonstrates an identical structural contraction: from step 25 to step 450, single-sample accuracy more than triples while solution coverage drops by 43%. This contraction directly mirrors a steep erosion in opening diversity: among valid GRPO solutions, entrance family coverage declines from 0.932 to 0.571, accompanied by a drop in entrance Shannon entropy from 1.585 to 0.760 nats (Table 19). This tradeoff is invariant to decoding configurations, persisting across varying generation temperatures, top-p thresholds, and maximum token budgets (Appendices C.5, C.6, C.7, and C.8).

Table 1: **Accuracy and solution breadth over two independently trained RLVR runs on Countdown**. PPO and GRPO use 150 and 135 held-out problems, respectively. Endpoints use 320 samples per problem and comparisons are within run. Intervals shown in Appendix C.9.

RLVR	Checkpoint	pass@1	pass@64	pass@256	Solution cov.@320	Operator-class cov.
PPO run	Base	0.001	0.076	0.228	0.097	0.178
	Step 25	0.005	0.218	0.416	0.189	0.354
	Step 50	0.051	0.500	0.596	0.337	0.538
	Step 75	0.129	0.482	0.543	0.283	0.502
	Step 150	0.323	0.441	0.457	0.152	0.362
	Step 275	0.285	0.376	0.385	0.111	0.253
Public GRPO run	Step 25	0.121	0.633	0.750	0.467	1.000
	Step 450	0.429	0.570	0.621	0.268	0.832

Breadth collapse persists on invariant solvable subsets. A potential confounding hypothesis is that coverage drops simply because the model forgets how to solve harder problems entirely. We rule this out by tracking solution leaf turnover on a fixed subset of 58 problems solved at both early (step 50) and late (step 275) PPO checkpoints (Appendix C.5). While 64% of problems solved at step 50

[1] Because the un-tuned base model generates $< 3\%$ format-valid outputs, step 50 serves as the earliest checkpoint exhibiting stable syntactic validity for reliable breadth benchmarking.

remain solvable at step 275, only 31% of the individual valid solution leaves discovered at step 50 are ever generated by the late policy (Appendix C.4). On this invariant solvable subset, mean coverage falls from 0.564 to 0.286 even as formatting compliance approaches 100%. Moreover, no problem missed at step 50 is newly unlocked at step 275 under 320-sample budget. Even at 2,048 samples per problem, 33 problems lost between steps 50 and 275 remain unrecovered (Appendix C.5). Therefore, RLVR does not broaden problem reachability; rather, it consolidates probability mass onto a narrow subset of previously accessible pathways while extinguishing alternative valid solutions.

Contraction concentrates disproportionately at initial computational branches. Stratifying the solution space into distinct granularities reveals that diversity loss is most severe at the earliest decision points. Among valid solutions, the coverage of coarse operator classes decreases by half over training, while coverage across the finer first-evaluated-pair partition drops by two thirds to 0.117 (Table 7, Appendix C.1). Tracking next-operator distributions across pre-operator states demonstrates that entropy contraction is maximal prior to the first arithmetic operation and attenuates at downstream calculation steps. This monotonic drop in opening diversity recurs across six independent solver-enumerated structural partitions (Appendix C.6), establishing that the policy's solution space narrows primarily at its opening branch points.

5 Mechanistic Localization: Breadth Is Lost at the Entrance

The contraction of an RLVR policy's solution space can stem from two distinct failure modes: an **access failure** (the policy fails to initiate a valid solution branch) or an **execution failure** (the policy initiates a branch but fails to carry out the necessary downstream arithmetic). If access dominates, three empirical signatures must hold: 1). Likelihood shifts during training must concentrate predominantly around the opening arithmetic decision. 2). Supplying an unselected entrance prefix should restore completion rates in low-access families. 3). Downstream execution capability on matched supplied entrances should remain stable or improve even as autonomous entrance access collapses. Table 2 summarizes these evaluations across both training pipelines.

Table 2: **Access and execution diagnostics on PPO and public GRPO Countdown.** The first row gives early-to-late increase in NLL per token in entrance segment vs. subsequent execution; the second compares designated-family completion with and without a supplied minimal entrance at late checkpoint; the third tracks minimal-entrance completion between early and late checkpoints.

Diagnostic	PPO	GRPO
NLL increase per token: entrance vs. execution	$+4.46$ vs. $+0.28$	$+8.08$ vs. $+0.71$
Designated completion: none $\rightarrow$ minimal entrance	$0.018 \rightarrow 0.212$	$0.126 \rightarrow 0.227$
Minimal-entrance completion: early $\rightarrow$ late	$0.113 \rightarrow 0.212$	$0.160 \rightarrow 0.227$

Likelihood shifts concentrate disproportionately prior to first operation. We isolate where probability mass migrates by computing teacher-forced negative log-likelihood (NLL) profiles over 1,392 paired solver-constructed valid continuations at PPO steps 25 and 275, splitting each trajectory at the boundary of the first arithmetic operator (Appendix D.3). Because teacher forcing does not require the model to generate a valid completion autonomously, step 25 serves as an early, unbiased baseline. Under PPO, the entrance segment experiences a substantial likelihood drop of 4.46 nats per token (95% CI $[4.16, 4.76]$), compared to a negligible shift of only 0.28 nats per token ($[0.25, 0.32]$) across the remaining reasoning steps, i.e., a $16\times$ disparity in divergence. Token-level attribution reveals that the largest single likelihood drop falls directly on the initial operand token that determines family selection. On 232 valid continuations under GRPO (steps 25 vs. 450), the entrance shift is 8.08 nats per token ($[7.96, 8.19]$) vs. 0.71 nats ($[0.58, 0.85]$) during execution, i.e., an $11\times$ ratio. Across both algorithms, RLVR updates suppress the probability of entering alternative solution paths rather than degrading the policy's capacity to express valid downstream arithmetic.

Minimal entrance prefixes restore completion in low-access families. To test whether unvisited solution families remain executable, we prompt checkpoints with solver-constructed prefixes of varying specificity appended to a neutral scaffold: (i) no entrance, (ii) a minimal entrance specifying only the first operand and operator (e.g., "`Let me try: 4 *`"), and (iii) a fully completed first calculation (e.g., "`4 * 7 = 28`"). We designate up to two feasible families per problem and measure rate of valid completions falling strictly inside designated family (Table 3, Panel A; Figure 2b). At PPO step 275, supplying a minimal entrance boosts designated-family completion by more than an order of magnitude, i.e., from 0.018 under free sampling to 0.212. The large recovery already

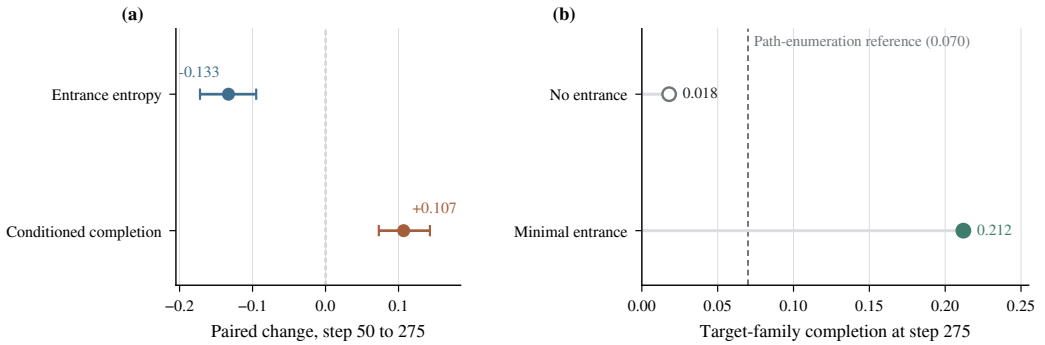

Figure 2: **Access and execution move in opposite directions (PPO run).** (a) From step 50 to step 275, normalized entrance entropy falls while completion from supplied entrances rises on matched problem and family keys; points are paired changes with 95% problem-bootstrap intervals. (b) At step 275, designated-family completion rises from 0.018 to 0.212 after a minimal entrance on the same 150 problems; the dashed line is the path-enumeration reference (Appendix D.1).

appears when only the operand and operator are supplied. Late GRPO checkpoint exhibits an identical recovery ($0.126 \rightarrow 0.227$). Furthermore, grafting a minimal entrance directly into model's own failed rollouts recovers a 0.176 valid completion rate (Panel B), whereas shorter non-arithmetic cues produce no recovery (Appendix D.2). Therefore, specifying a single opening operation unlocks latent solution families without requiring external hints regarding downstream logic.

Entrance access collapses while conditional execution capability improves. We conduct a paired evaluation across 139 problems and 359 feasible family instances by matching identical $(\text{problem}, \text{family}, \text{scaffold})$ tuples between PPO steps 50 and 275. As training proceeds, entrance access concentrates sharply: opening entropy drops while top-1 family rises on matched problem and family keys. In contrast, conditional execution capability on exact same supplied entrances improves significantly, rising by $+0.107$ (95% CI $[0.073, 0.143]$; Figure 2a; Appendix D.5). Crucially, this preservation extends even to completely extinct pathways: for 28 entrance families observed at step 50 but never sampled at step 275 across 320 free rollouts, clamping minimal entrance achieves a 0.529 designated completion rate at step 275 (Table 3, Panel C; Appendix D.6). Rather than suffering internal execution collapse, alternative reasoning paths remain latent and highly competent, i.e., the trained policy simply ceases to enter them.

Table 3: **Entrance interventions on the PPO Countdown run**. Panel A reports designated-family completion over all 150 problems, with 8 to 16 continuations per prefix and a 0.070 path-enumeration reference conditional on the entrance. In Panel B we graft the same entrances into failed traces at step 275 (S_{loss}: 33 problems, 32 with a usable retry point; reference 0.044). Panel C covers the 28 families observed at step 50 but unobserved in 320 free samples at step 275 with n=64 continuations per cell (the matched completed-calculation comparison is in Table 14, Panel D). The GRPO counterpart is explained in Appendix E.

		Step 50	Step 275	Step 275 any-valid
Panel A	No entrance	0.009	0.018	0.060
	Minimal entrance	0.113	0.212	0.261
Panel B	Feasible minimal entrance	-	0.176	0.227
	Infeasible entrance	-	0.000	0.069
	Empty graft	-	0.000	0.117
Panel C	All 28 families	0.475	0.529	0.584

6 Targeting Opening Decisions Recovers Solution Breadth

If reasoning breadth is predominantly lost at the entrance rather than during downstream execution, an immediate corollary follows: interventions that alter surface formatting or terminal answers should yield minimal recovery, whereas interventions that restore or redistribute early computational states should recover breadth in proportion to how effectively they reopen the initial entrance distribution. We evaluate fixed interventions across our saved PPO checkpoints on 100 held-out problems (Table 4); Appendix F gives the layer sweep and paired protocols.

Prompting and decoding changes yield modest coverage gains. Prompting the late policy (step 275) to generate a different valid method shifts coverage by $+0.014$ (95% CI $[0.000, 0.038]$). Blending step-50 logits during reasoning gives a similar gain of $+0.013$ $[0.004, 0.025]$. Clamping an alternative solver-feasible operator under the same first operand changes coverage by -0.007 (Appendix F.1), while answer-only mixing leaves coverage at 0.105. Raising temperature increases coverage by $+0.016$ $[0.007, 0.027]$, with pass@1 falling from 0.278 to 0.264.

Layer interpolation restores entrance diversity and solution breadth. Blending layers 20–28 of the late checkpoint with their step-50 weights increases solution coverage from 0.111 to 0.152 and entrance-family coverage from 0.185 to 0.251, with pass@1 of 0.305 versus 0.278 for the late policy. The entropy of generated reasoning entrances rises from 0.046 to 0.183 nats. Previously unobserved families also yield successful rollouts again (Appendix F.4). At the trajectory level, splitting a fixed 64-sample budget evenly between steps 50 and 275 yields the largest paired coverage gain ($+0.100$ $[0.059, 0.146]$), with single-sample accuracy shifting toward the early checkpoint. These results connect temporal checkpoint sampling and weight ensembling (Dang et al., 2025; Li et al., 2026c) to the recovery of opening exploration.

Table 4: **Paired interventions on 100 held-out problems**, with 64 samples per problem and a common step-275 control. Δ gives the paired change in solution coverage. † marks pass@1 equivalence under the ± 0.02 TOST margin; intervals appear in Table 22.

Intervention	pass@1	Solution cov.@64	Δ cov.	Entrance cov.@64
Step-275 policy (control)	0.278	0.111	—	0.185
Prompt diversification	$0.282^{\dagger}$	0.125	$+0.014$	0.201
Forced alternative operator	0.286	0.104	-0.007	0.174
Answer-only logit mixing	$0.269^{\dagger}$	0.105	-0.006	0.169
High temperature (T=1.0)	0.264	0.127	$+0.016$	0.212
Reasoning-phase logit mixing	$0.277^{\dagger}$	0.124	$+0.013$	0.209
Layer interpolation	0.305	0.152	$+0.041$	0.251
Checkpoint sampling	0.161	0.211	$+0.100$	0.315

Solver-guided entrance allocation recovers diversity without historical checkpoints. When earlier model checkpoints or weight averages are unavailable, test-time compute can be explicitly budgeted across feasible entrance families. Under a matched per-problem token cap, distributing rollouts uniformly across solver-identified feasible entrances outperforms standard free resampling by $+0.077$ (95% CI $[0.048, 0.109]$) in solution coverage and expands the number of distinct entrance families explored by $+0.430$ ($[0.297, 0.570]$) across the 128 problems containing non-trivial failure rollouts (Appendix F.3).[2] At step 50, this structured allocation yields no significant advantage over free sampling, directly confirming our theoretical prediction: explicit entrance budgeting becomes uniquely valuable precisely when RLVR optimization causes the policy's autonomous exploration distribution to collapse.

7 Generalization Across Model Scales, Horizons, and Training Pipelines

While Countdown enables exact decomposition of access and execution via exhaustive state-space enumeration, we now investigate whether early entrance narrowing generalizes to multi-step math reasoning benchmarks with unconstrained solution spaces, larger parameter scales (7B and 14B), and alternative post-training pipelines.

Early-step entropy collapse recurs across 7B and 14B model scales. Evaluating Qwen2.5 base and SimpleRL pairs (Zeng et al., 2025) across six standard math benchmarks demonstrates that RLVR consistently triggers early distributional collapse at scale (Table 36; Appendix H.1). While SimpleRL increases macro pass@1 by 13.1 percentage points at 7B and 16.9 at 14B, asymptotic coverage remains flat, i.e., pass@64 shifts negligibly from 0.754 to 0.760 at 7B and remains identical at 0.774 at 14B. Concurrently, first-calculation entropy drops by 30% at 7B and 41% at 14B, while distinct-trace rate nearly halves. Scaling rollout budget to 256 samples per problem on GSM8K and MATH500 confirms this saturation: repeated sampling yields diminishing marginal accuracy gains under SimpleRL because early computational diversity remains several-fold lower (Appendix G.3).

Same-trace likelihood attribution confirms early-stage divergence across math benchmarks. To localize where probability mass shifts in complex reasoning traces without alignment, we score identical reference solutions under both base and RLVR checkpoints, bifurcating per-token log-likelihoods at the boundary of the first complete calculation (Figure 3; Appendix G.1). Across all six

[2]The remaining 22 evaluation problems achieve a 100% native pass rate and explore their full feasible sets.

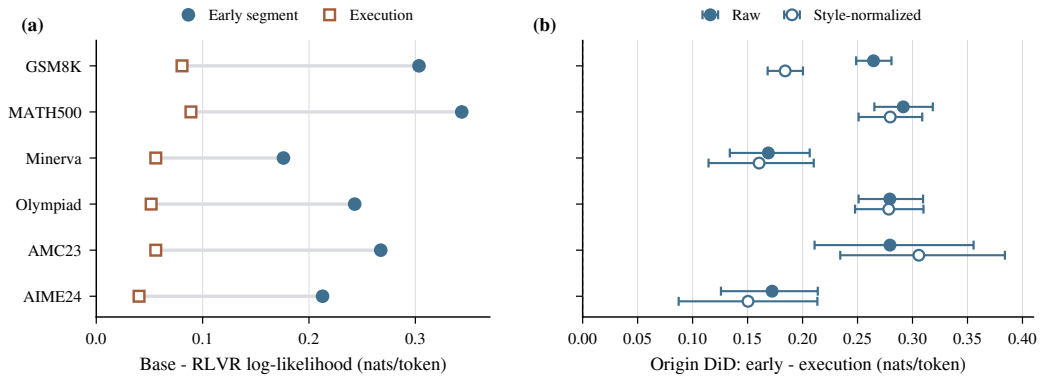

Figure 3: **Early concentration on standard math benchmarks.** (a) Same-trace scoring on Qwen2.5-7B: base minus RLVR per-token log-likelihood before the first complete calculation and during subsequent execution on Base-origin traces. (b) Origin-stratified difference-in-differences (DiD) on the same source traces before and after style normalization, with 95% problem-cluster bootstrap intervals (Eq. 7). Values and sample counts appear in Appendix G.1.

benchmarks, the origin-stratified early-versus-execution contrast is positive and remains positive after style normalization of matched source traces (Appendix G.1). Blinded semantic resegmentation resolves 19,160 of 19,941 evaluated traces (96.1%) and confirms that this early-segment divergence is robust across independent model-annotated boundaries (Appendix G.2). Therefore, the entrance-narrowing signature identified in Countdown operates as the primary driver of policy concentration across standard mathematical reasoning benchmarks.

Conditional recovery extends to multi-step mathematics. Supplying completed first calculations that the base policy reaches but the RLVR policy does not sample yields completion rates of 0.850 on GSM8K and 0.713 on MATH500 (Appendix G.4). These alternative routes remain usable once their initial arithmetic state is supplied.

Multi-stage alignment pipelines elevate accuracy while preserving search breadth. Solution space contraction is not an inescapable tax on reasoning capability; rather, it is sensitive to the optimization curriculum (Table 5). Across the OLMo-3 7B series (Team Olmo, 2025), pass@1 steadily increases across the SFT $\rightarrow$ DPO $\rightarrow$ RLVR stages, but first-calculation entropy and distinct-trace rates remain remarkably stable near their initial SFT baselines. Similarly, DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025) achieves both the highest single-sample accuracy (pass@1) and the highest first-calculation entropy among the evaluated Qwen 7B models (Table 5). A separate 1.5B comparison shows accuracy gains after RLVR on a distilled initialization with stable first-calculation structure (Appendix H.2).

Table 5: **Post-training comparison at 7B**, using unweighted macro averages of problem means over six benchmarks; the 14B pair comparison is shown in Appendix H.1. First-calculation entropy measures early-decision breadth and the distinct-trace rate is a length-sensitive complement.

Model	pass@1	pass@64	First-calc. H	Distinct-trace rate
Qwen Base 7B	0.379	0.754	1.063	0.780
Qwen SimpleRL 7B	0.510	0.760	0.744	0.444
Qwen Distill 7B	0.526	0.795	1.240	0.871
OLMo-3 7B (SFT)	0.447	0.781	1.189	0.770
OLMo-3 7B (DPO)	0.508	0.784	1.269	0.786
OLMo-3 7B (RLVR)	0.613	0.828	1.137	0.763

Trajectory-level case study: Direct RLVR prunes viable opening strategies. To ground benchmark-level entropy collapse in concrete reasoning dynamics, Table 6 details a trajectory-level case study on GSM8K Item 108, an instance where every evaluated checkpoint reaches ceiling capability (pass@64 =1.0). The problem naturally admits three distinct reasoning entrances: an intuitive backward deduction ($110 + 5 \rightarrow 115 - 15 \rightarrow 100/2$), a forward algebraic formulation ($(2x+15)-5 = 110$), and a simplified net-change relation ($2x = 110-10$). While Qwen-7B Base frequently accesses the backward deduction (Sample 6; 10 of 59 correct rollouts), **Qwen SimpleRL** (direct RLVR) completely extinguishes this pathway across all 64 rollouts ($10 \rightarrow 0$), concentrating 53 completions exclusively onto the forward algebraic setup (Sample 0). In contrast, the multi-stage **OLMo-3** alignment pipeline preserves the backward entrance throughout its entire training curriculum (SFT: 1, DPO: 8, RLVR: 6), with its final RLVR checkpoint (Sample 12) retaining all three entrance formulations while achieving a perfect $64/64$ success rate. Direct RLVR thus prunes vi-

Table 6: **Qualitative comparison of reasoning entrances on GSM8K Item 108.** All models reach pass@64 =1. Direct RLVR (**Qwen SimpleRL**) collapses onto a dominant forward equation (**red**), completely extinguishing the backward-arithmetic entrance ($10 \rightarrow 0$ out of 64 rollouts). In contrast, the staged alignment pipeline (**OLMo-3 RLVR**) preserves access to the backward deduction (**blue**). Trajectories are condensed and initial arithmetic commitments are colored; the full text of the selected rollouts appears in Appendix I.

Question	*Henry wants to bake twice as many cookies as last year. When he finishes baking, he realizes he actually baked 15 more cookies than he meant to. He drops 5 of his cookies as he is putting them out to cool, and now has a total of 110 cookies. How many cookies did Henry bake last year?* ***(Gold Answer: 50)***
Qwen Base 7B Sample 6; correct **Reverse Entrance**	Henry now has 110 cookies. He dropped 5 cookies, so before dropping them he had **110 + 5 = 115** cookies. He baked 15 more than he meant to, so he meant to bake 115 - 15 = 100 cookies. This is twice last year, so he baked 100 / 2 = **50** cookies last year. The answer is 50.
Qwen SimpleRL 7B Direct RLVR; Sample 0 **Forward Equation**	Let x be the number of cookies Henry baked last year. He intended to bake $2x$, so he baked $2x + 15$ cookies. After dropping 5, he has **(2x + 15) - 5 = 110** cookies left. Simplifying gives $2x + 10 = 110$, so $2x = 100$ and $x =$ **50**. The answer is 50.
OLMo-3 7B (RLVR) Staged RLVR; Sample 12 **Reverse Entrance**	Henry drops 5 cookies while cooling, and then has a total of 110 cookies left. So before dropping the 5 cookies, he had: **110 + 5 = 115** cookies. This 115 cookies is the actual number he baked: $2x + 15 = 115 \implies 2x = 115 - 15 = 100 \implies x = 100/2 =$ **50**. The answer is 50.

able opening strategies even on problems within full model competence, whereas staged alignment maintains entrance plurality into late training.

Corroboration and entrance preservation across open-domain benchmarks. This divergence is consistent across open-domain mathematical problems. On a second GSM8K instance (Item 392, streaming discount; pass@64 =1.0 for all models), direct RLVR concentrates 55 of 63 correct completions onto a single initial addition ($10 + 10 = 20$), causing first-step calculation entropy to collapse from 2.148 to 0.517 nats. Conversely, OLMo-3 RLVR sustains diverse initial operations, such as computing per-item savings (10×0.20) or discounted prices (10×0.80), maintaining a high first-calculation entropy of 2.429 nats. An identical preservation pattern recurs on MATH500 (Appendix I.3). Because open-domain reasoning spans rely on structural parsing rather than closed-form solver graphs, the full text of selected rollouts appears in Appendix I, with all 960 case-study rollouts in the supplementary data.

Supervised multi-solution fine-tuning retains broad solution support. To test whether diversity preservation can be explicitly engineered, we train supervised fine-tuning (SFT) models on Qwen2.5-3B using solver-enumerated multi-solution demonstrations across disjoint training instances, strictly holding out the 135 evaluation tasks by semantic key (Table 38; Appendix H.3). At pass@1 $= 0.333$, SFT maintains a solution coverage of 0.709, i.e., more than $2.6\times$ the coverage of the late GRPO checkpoint (0.268 coverage at pass@1 $= 0.429$). Furthermore, sweeping $k \in \{1, 2, 4, 8\}$ solutions per problem over 8,000 training problems reveals a strict monotonic dose–response: solution coverage scales from 0.471 at $k = 1$ to 0.766 at $k = 8$. Learning correct paths can therefore preserve broad solution support. Continuing GRPO from the k=4 SFT model concentrates training-response probabilities again (Appendix H.3).

8 Conclusion

RLVR-induced solution narrowing is fundamentally an *access* failure rather than an *execution* collapse. Decoupling opening branch selection from downstream computation on Countdown reveals that probability mass drains almost entirely at the entrance ($11\times$–$16\times$ larger likelihood shifts), while latent execution capability remains intact. Layer interpolation restores entrance diversity and increases solution coverage by 37%, with pass@1 of 0.305 versus 0.278 for the late policy; solver-guided entrance allocation also recovers breadth. This early-step collapse recurs at 7B and 14B scales across standard math benchmarks. However, multi-solution SFT and staged alignment pipelines demonstrate that the training pipeline shapes solution breadth as well as reasoning accuracy. Effective test-time scaling and training regularization must therefore target the initial decision boundary: scalable reasoning requires not only executing a chosen path to completion, but keeping the doors to alternative solutions open.

AI USE STATEMENT

The authors used generative AI tools for language editing, literature discovery, refinement of visualization code, and blinded annotation of first-calculation spans. The authors verified the resulting text, references, code, and figures and take full responsibility for the paper.

REPRODUCIBILITY STATEMENT

Protocols are described in §3 and the appendices, including entrance construction and its information checks, per-experiment sampling budgets, seeds, and the evaluation protocol for the public GRPO series. Appendix C.10 reports the independent enumerator cross-check. The enumerated Countdown solution sets, entrance-construction code, scoring code, and evaluation scripts can be accessed via `https://anonymous.4open.science/r/early-branch-locking-C0E1`.

REFERENCES

Eric J Bigelow, Ari Holtzman, Hidenori Tanaka, and Tomer Ullman. Forking paths in neural text generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8RCmNLeeXx.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024. URL https://arxiv.org/abs/2407.21787.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Xingyu Dang, Christina Baek, Kaiyue Wen, J Zico Kolter, and Aditi Raghunathan. Weight ensembling improves reasoning in language models. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=S2IKxulLT1.

Marius Dragoi, Ioana Pintilie, Florin Gogianu, and Florin Brad. Beyond pass@k: Breadth-depth metrics for reasoning boundaries, 2025. URL https://arxiv.org/abs/2510.08325.

Jingchu Gai, Guanning Zeng, Huaqing Zhang, and Aditi Raghunathan. Differential smoothing mitigates sharpening and improves llm reasoning, 2025. URL https://arxiv.org/abs/2511.19942.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin

Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning, 2025. ISSN 1476-4687. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting GRPO beyond distribution sharpening. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25548–25560, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1298. URL https://aclanthology.org/2025.emnlp-main.1298/.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zican Hu, Shilin Zhang, Yafu Li, Jianhao Yan, Xuyang Hu, Leyang Cui, Xiaoye Qu, Chunlin Chen, Yu Cheng, and Zhi Wang. Diversity-incentivized exploration for versatile reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9G7AbBrd27.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=QnjfkhrbYK.

Feng Ju, Zeyu Qin, Rui Min, Zhitao He, Lingpeng Kong, and Yi R. Fung. Reasoning path divergence: A new metric and curation strategy to unlock llm diverse thinking, 2026. URL https://arxiv.org/abs/2510.26122.

Soeun Kim and Albert No. Where rollouts begin: Low-load, high-leverage first-token diversification for rlvr, 2026. URL https://arxiv.org/abs/2605.28295.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=IFXTZERXdM7.

Long Li, Zhijian Zhou, JIARAN HAO, Jason Klein Liu, Yanting Miao, Wei Pang, Xiaoyu Tan, Wei Chu, Zhe Wang, Shirui Pan, Chao Qu, and Yuan Qi. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=xPEsxcO7F7.

Ruizhe Li, Chen Chen, Yuchen Hu, Yanjun Gao, Xi Wang, and Emine Yilmaz. Attributing response to context: A Jensen–Shannon divergence driven mechanistic study of context attribution in retrieval-augmented generation. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=7bHjHAOXH9.

Yuetai Li, Zhangchen Xu, Fengqing Jiang, Bhaskar Ramasubramanian, Luyao Niu, Bill Yuchen Lin, Xiang Yue, and Radha Poovendran. Temporal sampling for forgotten reasoning in LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28309–28327, San Diego, California, United States, July 2026c. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1305. URL https://aclanthology.org/2026.acl-long.1305/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. Are your LLMs capable of stable reasoning? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17594–17632, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.905. URL https://aclanthology.org/2025.findings-acl.905/.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=YPsJha5HXQ.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Uzay Macar, Paul C. Bogdan, Senthooran Rajamanoharan, and Neel Nanda. Thought branches: Interpreting LLM reasoning requires resampling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=bVsAuIOvJ5.

Kohsei Matsutani, Shota Takashiro, Gouki Minegishi, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. RL squeezes, SFT expands: A comparative study of reasoning LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=N2lMNqJsBw.

Phuc Minh Nguyen, Chinh D. La, Duy M. H. Nguyen, Nitesh V. Chawla, Binh T. Nguyen, and Khoa D. Doan. The reasoning boundary paradox: How reinforcement learning constrains language models, 2025. URL https://arxiv.org/abs/2510.02230.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. GitHub repository, 2025. URL https://github.com/Jiayi-Pan/TinyZero.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Soumadeep Saha, Krish Sharma, Akshay Chaturvedi, and Nicholas Asher. Bodhi: Do llms branch out and discover heterogeneous inferences?, 2026. URL https://arxiv.org/abs/2608.02867.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018. URL https://arxiv.org/abs/1506.02438.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Yuda Song, Julia Kempe, and Remi Munos. Outcome-based exploration for llm reasoning, 2025. URL https://arxiv.org/abs/2509.06941.

Team Olmo. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=yfcpdY4gMP.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jGbRWwIidy.

Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may or may not escape its origin, 2026. URL https://arxiv.org/abs/2507.14843.

Lecheng Yan, Ruizhe Li, Guanhua Chen, Qing Li, Jiahui Geng, Wenxi Li, Longyue Wang, and Chenyang Lyu. Spurious rewards paradox: Mechanistically understanding how RLVR activates memorization shortcuts in LLMs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=SGUSUm2491.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and KC Tan. Diversity-aware policy optimization for large language model reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=5eZ0iykpDU.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Yang Yu. Pass@k metric for rlvr: A diagnostic tool of exploration, but not an objective, 2025. URL https://arxiv.org/abs/2511.16231.

Suqin Yuan, Jinkun Chen, Jiyang Zheng, Muyang Li, Lei Feng, Dadong Wang, Tao Xiang, Tongliang Liu, and Bo An. Understanding diversity collapse in rlvr via the lens of overtraining, 2026. URL https://arxiv.org/abs/2606.15455.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=4OsgYD7em5.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun MA, and Junxian He. SimpleRL-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=vSMCBUgrQj.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. BREAD: Branched rollouts from expert anchors bridge SFT & RL for reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=NUDaln2vCe.

Rosie Zhao, Alexandru Meterez, Sham M. Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: RL post-training amplifies behaviors learned in pretraining. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=dp4KWuSDzj.

Todd Zhou. When rlvr shrinks the reasoning boundary: Diagnosing pass@k inversion, 2026. URL https://arxiv.org/abs/2607.20543.

Appendix: Overview & Table of Contents

CONTENTS

A Limitations **16**

B Full Setup: Entrances, Access, and Execution **16**

C Countdown Narrowing and Robustness (PPO) **18**

C.1 Opening partitions 18

C.2 On-policy pressure at entrances 18

C.3 Operator-class coverage over training 19

C.4 Leaf turnover 19

C.5 Survivorship control 19

C.6 Alternative partitions 20

C.7 Token-cap re-evaluation 20

C.8 Decoding sweep 21

C.9 Bootstrap intervals 21

C.10 Solver cross-check 22

D Entrance Protocol and Evidence (PPO) **22**

D.1 Prefix construction and information checks 22

D.2 Intermediate prefixes and controls 22

D.3 Teacher-forced likelihood 22

D.4 Failed-trace grafts 23

D.5 Paired access and execution 24

D.6 Access-tail recoverability 24

D.7 Successful-trace prefixes 25

E Replication on a Public GRPO Run **25**

E.1 Breadth at the selected endpoints 26

E.2 Teacher-forced likelihood 26

E.3 Supplied entrances 26

F Reopening Entrances (PPO) **27**

F.1 Forced operator override 28

F.2 Cross-run confirmation on public GRPO 28

F.3 Entrance allocation 28

F.4 Entrance recovery under interpolation 28

F.5 Intermediate-checkpoint comparison 29

F.6 Layer and mixing-weight sweep 30
F.7 Equal-length position interventions 30

G Math Benchmarks: Early Concentration and Its Limits **30**
G.1 Same-trace scoring 31
G.2 Semantic boundary resegmentation 31
G.3 Sampling depth at 256 samples 32
G.4 Conditional recovery from supplied arithmetic states 33

H Scale, Pipelines, and Supervised Controls **33**
H.1 Scale comparison 33
H.2 RL after distillation 33
H.3 Supervised fine-tuning controls on Countdown 34

I Raw Trajectories for Qualitative Case Studies **35**
I.1 GSM8K Item 108 (Baking Cookies, Gold Answer: 50) 35
I.2 GSM8K Item 392 (Streaming Bundle, Gold Answer: 34) 37
I.3 Companion Analysis on MATH500 (Item 451) 39

J Training and Evaluation Details **43**

K Extended Related Work **43**

A Limitations

Exhaustive enumeration is what makes access and execution separately measurable, which currently ties the direct analysis to Countdown. Our PPO run uses rollout group size $G{=}1$, no actor KL penalty, and a 1,024-token training rollout cap; the second run is a public GRPO series we evaluate rather than train; and SFT controls use LoRA fine-tuning. On standard math benchmarks the first complete calculation is an observable proxy for an entrance, not a solver-defined one. The supplied-entrance quantity E_t^{do} is interventional: it measures completion after an entrance is set externally, whereas the execution term in Equation 2 conditions on entrances selected by the policy itself.

B Full Setup: Entrances, Access, and Execution

Task formulation and training regimes. A Countdown instance (Pan et al., 2025) specifies a target integer and a set of three or four input operands. A valid solution is an arithmetic expression that evaluates exactly to the target using each given number precisely once with valid operations ($\{+, -, \times, \div\}$) and parentheses. To ensure algorithmic generality, we examine two independent RLVR training pipelines: **Self-Trained PPO:** Following the TinyZero framework (Pan et al., 2025), we train Qwen2.5-3B (Qwen et al., 2025) with PPO, evaluating the base model alongside eleven actor checkpoints saved from step 25 to step 275. **Public GRPO Checkpoints:** We evaluate the open-source `Qwen-2.5-3B-R1-countdown` series (Qwen2.5-3B-Instruct trained with TRL GRPO). To prevent evaluation leakage, we filter the public 50,000-example training superset against sorted input-target semantic keys, yielding a strictly disjoint evaluation set of 135 solver-feasible problems. An independent 300-problem validation sweep selects step 25 as the earliest checkpoint achieving

native-format validity above 90%, while step 450 serves as the late endpoint. Both setups maintain their native prompt templates and formatting conventions. Because the valid state space of Countdown is exhaustively enumerable, we can trace the precise trajectory of disappearing solutions and monitor how probability mass migrates across the policy distribution.

Solution space and coverage. For any feasible instance x, an exact symbolic solver generates the complete set of valid arithmetic expressions, canonicalized modulo commutativity and associativity for $+$ and $\times$. We formalize this search space as a decision tree where depth-one branches represent opening arithmetic decisions and terminal leaves constitute the set of canonical valid solutions $\mathcal{S}(x)$. Given n independent model rollouts $Y_{1:n}$, the per-instance solution coverage $\mathrm{Cov}(x, Y_{1:n})$ and corpus-level mean coverage $\overline{\mathrm{Cov}}(X)$ are defined as:

$$\mathrm{Cov}(x, Y_{1:n}) = \frac{\left|\mathcal{S}(x) \cap \mathrm{canon}(Y_{1:n})\right|}{|\mathcal{S}(x)|}, \quad \overline{\mathrm{Cov}}(X) = \frac{1}{|X|} \sum_{x \in X} \mathrm{Cov}(x, Y_{1:n}). \tag{3}$$

where $\mathrm{canon}(Y_{1:n})$ maps outputs to their canonical forms. An independent enumerator verifies problem feasibility and leaf-set completeness on 500 held-out instances with zero discrepancy (Appendix C.10).

Entrance families: Decoupling access and execution. We stratify the initial decision space into two granularities: a coarse partition by the first arithmetic operator (**operator class**) and a fine-grained partition by the initial operand-operator tuple $(o_1, \odot_1)$, which we term the **entrance family** b. For instance, given the input $\{4, 5, 7, 9\} \to 24$, feasible canonical entrance families include $5-$, $7-$, $4\times$, and $9-$ (Figure 1). The solver orders and renders each canonical expression before assigning its first operand and operator. For example, $(7 - 4) \times 5 + 9$ supplies the $7-$ family. Across 150 held-out test problems, the solver discovers 379 feasible families (averaging 2.53 families per instance). Because an entrance family fixes only the initial operation while leaving all subsequent arithmetic unconstrained, solving a problem through family b factorizes into an *access* phase and an *execution* phase:

$$\pi_\theta(\text{solve via } b \mid x) = \underbrace{\pi_\theta(B = b \mid x)}_{\textit{access}} \cdot \underbrace{\pi_\theta(\text{valid completion in } b \mid x, B = b)}_{\textit{execution}}. \tag{4}$$

Rollout entrance assignment. Let $B = g(Y)$, where $g(y)$ returns the first parseable operand–operator pair in the generated reasoning block, with $g(y) = \perp$ when none is found. Access is the frequency of $g(y) = b$ among all rollouts, including incorrect and malformed outputs. A successful rollout through b both solves the task and starts its final `<answer>` expression with b; the answer parser preserves generation order. Thus observational execution is the proportion of such successes among rollouts with $g(y) = b$. Canonical-family coverage instead assigns each correct expression to the family of its canonical representative. The two measurements retain, respectively, the model's generated choice and the solver's disjoint leaf partition.

We estimate family access $A_t(b \mid x)$ through unconstrained sampling and teacher-forced prefix log-probabilities. We directly measure conditional execution capability $E_t^{\mathrm{do}}(b \mid x)$ by clamping generation to a solver-constructed entrance prefix e_b: $E_t^{\mathrm{do}}(b \mid x) = \pi_t\big(\text{valid completion in family } b \,\big|\, x \oplus e_b\big)$. We use the superscript do to distinguish it from the observational execution term in Equation 4, which conditions on entrances selected by the policy itself. Throughout this paper, *designated-family completion* refers to conditional probability of reaching a valid solution strictly inside the clamped family b, whereas *any-valid success* denotes generating a valid solution in any family following the prefix.

Supplied entrance design and information control. To probe downstream execution without introducing confounding reasoning cues, every entrance prefix e_b is synthetically constructed by the solver rather than extracted from model-generated traces (e.g., "`Let me try: 4 *`"). Each prefix terminates immediately after the opening operator, enforcing the family constraint while leaving all downstream degrees of freedom entirely unguided. Control baselines, including neutral problem restatements and model-generated failed prefixes, verify baseline completion, while teacher-forced scoring evaluates full valid continuations. For fixed x and b, the template is identical across solver leaves and contains only the first operand and operator. Appendix D.1 gives the construction checks, and Appendix D.7 compares successful-trace prefixes.

On-policy gradient dynamics on entrance selection. Policy-gradient algorithms (e.g., PPO, GRPO) update policy parameters exclusively along sampled trajectories, allocating gradient updates

proportional to empirical visitation frequency. Consequently, an entrance family that underperforms relative to the moving baseline during initial iterations suffers an immediate drop in selection probability. Because rarely sampled entrances receive diminishing gradient signals, initial sampling asymmetries compound over training, driving a self-reinforcing contraction of the policy's opening support (Yuan et al., 2026; Zhou, 2026). We formalize this via a first-order entrance surrogate in Appendix C.2, which derives the feedback on entrance selection for fixed conditional execution rates.

Structural proxies for non-enumerable math benchmarks. Because standard multi-step mathematical benchmarks lack an exhaustively enumerable solution space $\mathcal{S}(x)$, we construct structural proxies over generated reasoning traces. A *reasoning trace* is defined as the canonicalized sequence of intermediate calculation steps in a rollout, normalized by stripping formatting whitespace and sorting commutative arguments. The *distinct-trace rate* measures the empirical diversity of these sequences. To capture opening diversity independent of reasoning length, we define the *first calculation* as the earliest parsed token span evaluating an instantiated arithmetic relation; its categorical entropy serves as our primary metric for early-stage search breadth. Furthermore, by evaluating identical reference traces across multiple checkpoint pairs, we bifurcate per-token log-likelihoods at the boundary of the first complete calculation, directly isolating early-step distributional shifts from downstream arithmetic execution.

Evaluation protocol and statistical rigor. For Countdown, PPO evaluations are conducted using temperature $T = 0.7$, top-$p = 0.9$, a maximum generation budget of 256 tokens, and 320 independent rollouts across 150 held-out solver-feasible problems disjoint from the training set. GRPO evaluations use identical decoding hyperparameters ($T = 0.7$, top-$p = 0.9$) with a 1,024-token cap, 320 rollouts, and the 135 filtered instances. For scaling and general mathematical reasoning analyses, we evaluate Qwen2.5-7B/14B Base and SimpleRL checkpoints (Zeng et al., 2025), DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025), and the full OLMo-3 7B alignment progression (SFT, DPO, and RLVR) (Team Olmo, 2025). The standard math suite spans GSM8K (Cobbe et al., 2021) (500 problems), MATH500 (Hendrycks et al., 2021; Lightman et al., 2024) (500), Minerva Math (Lewkowycz et al., 2022) (272), OlympiadBench (He et al., 2024) (500), AMC23 (40), and AIME24 (30). Rollouts for math benchmarks are generated with 64 samples per instance at $T = 0.6$, top-$p = 0.95$, and a 16,000-token cap; truncated outputs are scored as incorrect. Accuracy is reported via the standard unbiased pass@k estimator (Chen et al., 2021), with entropy computed in nats. All confidence intervals are estimated via bootstrap resampling (2,000 iterations for Countdown, 1,000 for benchmark structural statistics, and 10,000 for paired problem-cluster contrasts).

C Countdown Narrowing and Robustness (PPO)

This appendix collects the structural and robustness evidence behind §4. The central comparisons use step 50, the earliest format-reliable checkpoint, and step 275, the late endpoint.

C.1 Opening partitions

Table 7 expands the early-structure measurements of §4. Panel A measures access to feasible opening partitions at two resolutions. Panel B shows where operator probability moves at matched model-generated states immediately before an arithmetic operator, and Panel C measures operator entropy at solver-constructed canonical prefixes along the expression. Panel D pairs output-format validity with solution breadth.

C.2 On-policy pressure at entrances

Let $q_b = \pi_\theta(B = b \mid x)$ be access to family b and μ_b the verifier success conditional on entering b from self-generated states. With G rollouts, the probability that a training group contains a rewarded completion through b is

$$p_b^+(G) = 1 - (1 - q_b\mu_b)^G. \tag{5}$$

For $q_b \ll 1/G$, rewarded exposure scales as $Gq_b\mu_b$, and at $G{=}1$ directly as $q_b\mu_b$. For the local surrogate $J_x = \sum_b q_b\mu_b$ with $q = \mathrm{softmax}(u)$ and fixed μ_b,

$$\frac{\partial J_x}{\partial u_b} = q_b(\mu_b - \bar{\mu}), \tag{6}$$

Table 7: Countdown opening-partition diagnostics (150 problems, 320 samples per problem). Panel B conditions on matched model-generated pre-operator prefixes and Panel C on solver-constructed canonical prefixes.

	Step 50	Step 275
Panel A: access under two opening partitions among correct samples		
Operator classes per problem (feasible)	1.71	1.71
Coverage of feasible operator classes	0.538	0.253
First-evaluated-pair labels per problem (feasible)	3.99	3.99
Coverage of feasible first-evaluated pairs	0.368	0.117
Panel B: operator probability at matched model-generated pre-operator prefixes		
Addition	0.812	0.819
Subtraction	0.088	0.156
Division	0.094	0.025
Multiplication	0.006	0.001
Next-token operator entropy	0.574	0.415
Panel C: next-operator entropy at solver-constructed canonical prefixes (step 275)		
Before op_1		1.020
Before op_2		0.555
Before op_3		0.479
Panel D: output-format validity and solution breadth		
Answer-parse rate	0.240	0.290
Tag-order validity	0.954	0.997
Solution coverage@320	0.337	0.111

where $\bar{\mu} = \sum_j q_j \mu_j$ is mean conditional success under the current access distribution. The gradient scales with current access, so once a successful family becomes rare it is correspondingly less likely to supply further positive evidence. Equation 6 motivates the paired measurements of §4.

C.3 Operator-class coverage over training

Opening breadth declines gradually across checkpoints (Table 8), so the endpoint comparison is not a selection artifact.

Table 8: Coverage of feasible operator classes among correct samples, base model and all eleven RLVR checkpoints (150 problems, 320 samples).

Checkpoint	Base	25	50	75	100	125	150	175	200	225	250	275
Coverage	0.178	0.354	0.538	0.502	0.448	0.412	0.362	0.328	0.300	0.262	0.256	0.253

C.4 Leaf turnover

For each problem solved at step 50 we track which step-50 canonical leaves reappear later. Retention falls steadily while most of the same problems remain solvable: by step 275, 31% of step-50 leaves are observed again and 64% of step-50-solved problems are still solved.

C.5 Survivorship control

We split the 150 problems into S_{both} (58 solved at both steps 50 and 275) and S_{loss} (33 solved only at step 50); no problem is solved only at step 275 under the common budget. On S_{both}, solution coverage falls from 0.564 to 0.286 and operator-class coverage from 0.940 to 0.654, so the contraction holds within problems reachable at both endpoints. On S_{loss} the step-275 answer-parse rate is zero, which mixes format and access failures, so S_{both} carries the within-problem breadth comparison and the supplied-entrance experiments carry the identification. A budget stress test resamples the 33 S_{loss} problems and 33 matched S_{both} controls at step 275 with up to 2,048

Table 9: Retention of step-50 correct leaves at later checkpoints, 91 step-50-solved problems, 320 samples, 2,000 problem-bootstrap draws. "Zero retention" is the fraction of problems retaining no step-50 leaf.

Checkpoint	Leaf retention	Zero-retention fraction	Still solved
Step 75	0.722 [0.644, 0.789]	0.132	0.879
Step 100	0.644 [0.570, 0.717]	0.176	0.824
Step 125	0.584 [0.508, 0.656]	0.198	0.802
Step 150	0.476 [0.399, 0.547]	0.264	0.758
Step 175	0.432 [0.355, 0.506]	0.319	0.692
Step 200	0.380 [0.306, 0.452]	0.319	0.681
Step 225	0.322 [0.257, 0.390]	0.374	0.637
Step 250	0.306 [0.243, 0.371]	0.385	0.637
Step 275	0.310 [0.246, 0.378]	0.374 [0.275, 0.473]	0.637 [0.538, 0.736]

samples per problem: every S_{loss} problem remains unrecovered at every budget, while all 33 controls are recovered within 64 samples.

Table 10: Survivorship control, 320 samples per problem. Coverage columns are computed over correct samples.

Set (n)	Ckpt.	pass@1	pass@64	Format-valid	Solution cov.	Operator-class cov.
S_{both} (58)	50	0.117	0.960	0.331	0.564	0.940
S_{both} (58)	275	0.737	0.971	0.749	0.286	0.654
S_{loss} (33)	50	0.027	0.586	0.228	0.542	0.795
S_{loss} (33)	275	0.000	0.000	0.000	0.000	0.000

C.6 Alternative partitions

Coverage falls under all six solver-enumerated partitions of the same solution set (Table 11), and all endpoint intervals are disjoint. Generation-order operator and tree-root operator are genuinely different partitions, agreeing on only 0.218 and 0.265 of correct samples at the two endpoints.

Table 11: Coverage under solver-enumerated partitions among correct samples, 150 problems, 320 samples, 2,000 bootstrap draws.

Partition	Feasible labels/problem	Cov. step 50	Cov. step 275	Relative drop
Operator class (canonical)	1.71	0.538	0.253	0.53
Tree root operator	1.79	0.548	0.249	0.55
Full operator sequence	3.08	0.424	0.163	0.62
Tree signature	3.75	0.410	0.159	0.61
First evaluated pair	3.99	0.368	0.117	0.68
First merged numbers	2.49	0.483	0.167	0.66

C.7 Token-cap re-evaluation

We re-tokenize and re-evaluate all 96,000 endpoint samples under fixed response caps. At the 256-token cap, pass rates match the untruncated evaluation: 0.0508 and 0.2816 at steps 50 and 275, against 0.0513 and 0.2851. A 128-token cap truncates many answers before the closing tags and lies below the operating regime used here.

C.8 DECODING SWEEP

We sweep temperature, top-p, and min-p over $5 \times 2 \times 2$ configurations at step 275 (Table 12). Higher temperature recovers some breadth, and no setting reaches the step-50 coverage of 0.337. The best late-policy coverage is 0.194 at T=2.0, top-p=1.0, min-p=0.05, where pass@1 falls to 0.238.

Table 12: Decoding sweep at step 275, 150 problems and 320 samples per cell.

T	top-p	min-p	pass@1	pass@64	Solution coverage
0.7	0.90	0.00	0.284	0.377	0.108
0.7	0.90	0.05	0.283	0.376	0.108
0.7	1.00	0.00	0.277	0.406	0.138
0.7	1.00	0.05	0.283	0.385	0.109
1.0	0.90	0.00	0.274	0.402	0.130
1.0	0.90	0.05	0.274	0.402	0.138
1.0	1.00	0.00	0.267	0.432	0.149
1.0	1.00	0.05	0.275	0.408	0.131
1.3	0.90	0.00	0.252	0.425	0.157
1.3	0.90	0.05	0.264	0.417	0.151
1.3	1.00	0.00	0.222	0.445	0.169
1.3	1.00	0.05	0.267	0.434	0.149
1.6	0.90	0.00	0.154	0.438	0.158
1.6	0.90	0.05	0.260	0.438	0.170
1.6	1.00	0.00	0.107	0.441	0.184
1.6	1.00	0.05	0.256	0.448	0.165
2.0	0.90	0.00	0.034	0.401	0.144
2.0	0.90	0.05	0.239	0.446	0.174
2.0	1.00	0.00	0.009	0.295	0.147
2.0	1.00	0.05	0.238	0.467	0.194

C.9 BOOTSTRAP INTERVALS

Table 13 provides the complete trajectory of point estimates and 95% confidence intervals for the PPO Countdown run summarized in Table 1, evaluated across all 150 held-out problems with 320 rollouts per instance. All confidence intervals are computed via problem-cluster bootstrap with 2,000 resample draws (seed 1729) to account for problem-level variance. Crucially, the 95% confidence intervals for solution coverage at the earliest format-competent checkpoint (Step 50: $[0.277, 0.397]$) and the late checkpoint (Step 275: $[0.079, 0.145]$) are strictly disjoint, confirming that the two-thirds collapse in solution space breadth is statistically significant and not an artifact of rollout sampling noise. Between the two endpoints, single-sample accuracy rises while pass@64 falls from 0.500 to 0.376, accompanying the contraction in solution coverage.

Table 13: Problem-bootstrap intervals for the PPO Countdown run, 150 problems and 320 samples per problem, 2,000 draws, seed 1729.

Checkpoint	pass@1	pass@64	Solution coverage@320
Base	0.001 $[0.001, 0.002]$	0.076 $[0.054, 0.099]$	0.097 $[0.063, 0.135]$
Step 25	0.005 $[0.004, 0.006]$	0.218 $[0.174, 0.263]$	0.189 $[0.145, 0.235]$
Step 50	0.051 $[0.040, 0.062]$	0.500 $[0.431, 0.579]$	0.337 $[0.277, 0.397]$
Step 75	0.129 $[0.100, 0.159]$	0.482 $[0.407, 0.561]$	0.283 $[0.230, 0.337]$
Step 100	0.178 $[0.134, 0.224]$	0.446 $[0.373, 0.525]$	0.229 $[0.181, 0.278]$
Step 125	0.276 $[0.214, 0.340]$	0.446 $[0.373, 0.526]$	0.205 $[0.156, 0.254]$
Step 150	0.323 $[0.256, 0.393]$	0.441 $[0.365, 0.521]$	0.152 $[0.116, 0.190]$
Step 175	0.288 $[0.223, 0.357]$	0.399 $[0.324, 0.480]$	0.146 $[0.109, 0.187]$
Step 200	0.297 $[0.231, 0.366]$	0.401 $[0.327, 0.480]$	0.122 $[0.088, 0.160]$
Step 225	0.247 $[0.184, 0.312]$	0.382 $[0.309, 0.459]$	0.113 $[0.081, 0.148]$
Step 250	0.247 $[0.186, 0.311]$	0.381 $[0.307, 0.458]$	0.110 $[0.079, 0.144]$
Step 275	0.285 $[0.217, 0.351]$	0.376 $[0.303, 0.453]$	0.111 $[0.079, 0.145]$

C.10 Solver cross-check

A second enumerator shares no code with the primary solver, using ordered operand pairs and tuple-tree serialization instead of abstract-syntax-tree canonicalization. The two implementations agree on feasibility and on the complete canonical leaf set for all 500 held-out instances.

D Entrance Protocol and Evidence (PPO)

This appendix gives the construction, controls, and matched analyses behind §5.

D.1 Prefix construction and information checks

Each instance combines a scaffold with a solver-constructed prefix. The *neutral* scaffold restates the numbers and target in the model's normal format, and the *retry* scaffold places the prefix after an unsuccessful attempt. The conditions are:

- **No entrance**: no arithmetic entrance is supplied;
- **Minimal entrance**: the designated family's first operand and operator after "`Let me try:`";
- **Completed first calculation**: the first calculation with its intermediate value.

Designated families are taken from the feasible set in decreasing family size, up to two per problem; of the 150 problems, 45 contribute one family and 105 contribute two. Designated-family completion is scored from the first operand and operator in the final `<answer>` expression, preserving its written order. Each prefix receives 8 to 16 continuations at temperature 0.7. Rates are averaged within each problem before averaging across problems.

The minimal-prefix comparison uses all 150 problems. For the complete-calculation comparison, a supplied calculation must equal the witness expression's first evaluated binary operation, with an exact rational result. The resulting 127 problems are paired across all three conditions and both checkpoints (Table 14, Panel D).

Three reference quantities differ in role. The *path-enumeration reference*, 0.070 on average, is the fraction of enumerated arithmetic paths that reach the target once the designated operand and operator are fixed. The *leaf-uniform family share*, 0.396, is the fraction of canonical leaves belonging to the designated family. The unprompted designated-family completion rate at step 275 is 0.018.

For a fixed problem and entrance family, every solver leaf receives the same minimal template, `Let me try:` followed by the first operand and operator. All 510 scaffold–family prefixes satisfy this rule. The second operand, intermediate result, and later operations are supplied by the model. The auxiliary teacher-forced check compares family-witness likelihoods to measure whether a prefix identifies its designated family; the within-family information control is the template's invariance across leaves.

On the four-number intersection of 82 problems, step-275 minimal-entrance completion is 0.049 against a 0.013 path-enumeration reference, and 0.015 at step 50. Holding problem, family, and retry scaffold fixed, completion on this subset rises from 0.012 at step 25 to 0.044 at step 100 and 0.093 at step 200, so the effect is not specific to one late checkpoint. Six semantically equivalent minimal-entrance templates yield designated completion between 0.269 and 0.350, so the recovery effect does not depend on the surface wording of the cue.

D.2 Intermediate prefixes and controls

To find the smallest useful prefix we vary what is added at the retry point (Table 15). Generic retry text, the first number alone, and generic or misleading plans stay at baseline, while a completed local calculation produces a large effect. The useful intervention therefore sits between a bare cue and a downstream plan: it fixes one concrete arithmetic action and leaves the rest open.

D.3 Teacher-forced likelihood

For each instance we render a solver-constructed valid continuation and teacher-force the same continuation at steps 25 and 275, splitting the target at its first arithmetic operator. The analysis uses

Table 14: **Entrance interventions on the PPO Countdown run**. Panel A reports designated-family completion over all 150 problems, with 8 to 16 continuations per prefix and a 0.070 path-enumeration reference conditional on the entrance. In Panel B we graft the same entrances into failed traces at step 275 (S_{loss}: 33 problems, 32 with a usable retry point; reference 0.044). Panel C covers the 28 families observed at step 50 but unobserved in 320 free samples at step 275 with n=64 continuations per cell. Panel D compares the three prefix conditions on 127 matched problems. The GRPO counterpart is explained in Appendix E.

	Step 50	Step 275	Step 275 any-valid
Panel A: neutral scaffold, completion in the designated family			
No entrance	0.009	0.018	0.060
Minimal entrance (operand and operator)	0.113	0.212	0.261
Panel B: grafts into the model's own failed traces (step 275)			
Feasible minimal entrance	0.176 (reference 0.044; excess $+0.121$ $[0.047, 0.196]$)		0.227
Infeasible entrance		0.000	0.069
Empty graft		0.000	0.117
Panel C: families observed at step 50 but unobserved in 320 free samples at step 275 (28 families, n=64)			
All 28 families	0.475 $[0.353, 0.597]$	0.529 $[0.354, 0.697]$	0.584
paired $\Delta_{275-50} = +0.054$ $[-0.080, 0.190]$; vs. retained families at step 275: $\Delta = -0.129$ $[-0.375, 0.126]$			
Panel D: matched complete-calculation comparison (127 problems, neutral scaffold)			
No entrance	0.011	0.021	0.070
Minimal entrance	0.136	0.257	0.313
Completed first calculation	0.121	0.329	0.395

Table 15: Intermediate conditions on the retry scaffold, designated-family completion over 139 problems.

Checkpoint	Retry cue	First number	Local calc.	Generic plan	Misleading plan
Step 50	0.031	0.037	0.147	0.024	0.040
Step 275	0.015	0.015	0.211	0.023	0.004

1,392 prefixes with solver-consistent supplied states across 150 problems. We average prefix differences within each problem, then bootstrap problems with 10,000 draws (seed 1729). The entrance segment loses 4.464 nats per token $[4.158, 4.763]$, against 0.282 $[0.247, 0.319]$ during subsequent execution. In total NLL, the changes are 8.472 $[7.973, 8.959]$ and 3.034 $[2.635, 3.446]$. Teacher forcing scores step 25 directly, while step 50 remains the free-sampling reference.

Table 16 aligns the token-profile cache to the first arithmetic operator in the actual prompt–continuation encoding. Position 0 contains that operator. For example, the encoded sequence

Position	-2	-1	0	$+1$	$+2$	$+3$	$+4$
Token	3	6	␣−	␣	8	5	␣+

places both initial digits before the operator. Shorter initial operands account for the smaller support at position -2. The largest mean increase is on the initial operand, preceding the operator.

D.4 Failed-trace grafts

To test the entrance inside model-generated context we build grafts from step-275 traces that have already made a verified unsuccessful attempt. At the retry boundary we append a solver entrance for a feasible family, a solver-verified infeasible entrance, or no arithmetic state.

On S_{loss}, 32 of 33 problems admit a usable retry point. The feasible condition holds 84 graft instances: designated-family completion is 0.176 $[0.093, 0.263]$, any-valid success is 0.227 $[0.122, 0.338]$, and the problem-averaged excess over the matched path-enumeration reference is

Table 16: Teacher-forced NLL per token, aligned to the token containing the first arithmetic operator. Values average within problems before the problem-cluster bootstrap (10,000 draws). The paired profile covers 1,392 prefixes; position -2 has 1,258 tokens from 146 problems, and the other positions cover all 150 problems.

Relative position	Step 25	Step 275	Δ	Tokens
-2	$9.295\ [8.811, 9.763]$	$17.142\ [16.124, 18.123]$	$+7.847$	1258
-1	$1.951\ [1.610, 2.319]$	$3.490\ [2.865, 4.164]$	$+1.539$	1392
$+0$	$2.679\ [2.521, 2.843]$	$3.496\ [3.168, 3.833]$	$+0.816$	1392
$+1$	$0.741\ [0.620, 0.874]$	$0.876\ [0.676, 1.096]$	$+0.135$	1392
$+2$	$0.849\ [0.773, 0.927]$	$1.370\ [1.197, 1.553]$	$+0.521$	1392

$+0.121\ [0.047, 0.196]$. Completion rates and per-instance excesses are averaged within each problem, then across problems; the 0.044 reference is averaged over graft instances. The infeasible condition holds 64 instances with zero designated completion and 0.069 any-valid success, and the empty graft gives zero designated completion with 0.117 any-valid. A neutral-scaffold replication on S_{loss} yields excess 0.062 $[0.006, 0.132]$. The minimal entrance therefore reopens a feasible family inside the late policy's own failed reasoning, not only from a clean prompt.

D.5 Paired access and execution

Matching steps 50 and 275 on identical (problem, family, scaffold) keys gives 139 problems and 359 feasible family instances. Because per-family access shares sum to one within the designated set, access concentration is summarized at the set level. From step 50 to step 275, problem-paired entrance entropy falls by $0.144\ [0.104, 0.186]$ and normalized entropy by $0.133\ [0.095, 0.172]$, top-1 entrance share rises by $0.074\ [0.051, 0.098]$, and Gini rises by $0.062\ [0.042, 0.082]$. Mean absolute per-family access change is $0.115\ [0.098, 0.134]$.

On the same supplied entrances, designated-family completion improves by $+0.107\ [0.073, 0.143]$. Among instances whose step-275 access is below 0.05, mean completion is 0.216 against a mean path-enumeration reference of 0.054; for the rest it is 0.313 against 0.068. This pairing is the central contrast: training concentrates access over entrance families while improving completion from matched supplied entrances.

The free-generation traces also permit an observational decomposition of the change in solve probability. For each feasible family we estimate the execution term of Equation 2 from rollouts that enter the family without intervention and apply the exact two-factor Shapley identity to the change between steps 50 and 275. Aggregated over the 150 problems, the decomposition assigns 34.5% of the change in solve probability to access reallocation and 65.5% to conditional execution, exact up to machine precision. Access contracts the support over which solutions are sampled, while improved conditional execution supplies most of the increase in solve probability.

D.6 Access-tail recoverability

We call an entrance family *unobserved* when its access probability is at least 0.05 at step 50 and zero across 320 free samples at step 275 (28 families), and *retained* when access remains at least 0.05 at both checkpoints (24 families). Supplying the minimal entrance to the unobserved families yields designated-family completion $0.529\ [0.354, 0.697]$ at step 275 and $0.475\ [0.353, 0.597]$ at step 50, well above their mean path-enumeration reference of 0.115; retained families reach 0.658 $[0.464, 0.831]$ at step 275.

Recoverability varies across the access tail. On a separate 800-problem train-disjoint pool, a broader set of 276 zero-access family cells spanning 250 problems achieves mean supplied-entrance completion 0.087 $[0.057, 0.120]$. Conditional recoverability is therefore graded rather than binary: branches that once carried appreciable policy mass reopen readily, while the deeper tail is harder to reach from the same cue.

The zero-access designation itself is stable under larger budgets. On 150 held-out problems we draw 1,024 late-policy samples and split them into two independent halves. Of 337 families absent

from the first 512 samples, 336 remain absent from the second 512, none exceeds 0.02 access in the held-out half, and a zero hit gives a 95% Clopper–Pearson upper bound of 0.0058.

D.7 SUCCESSFUL-TRACE PREFIXES

An alternative is to cut prefixes from the model's own successful reasoning just before or after its first answer operator. These produce high downstream completion and also carry the remaining plan. At step 275 they yield completion 0.617 before the first operator and 0.999 after it, and their reasoning text contains intermediate operations and values even when the final expression is removed. The minimal solver template instead fixes only the operand and operator. Table 17 contrasts the two.

Table 17: Successful-trace prefixes and solver-constructed controls. Panel A is a state-only ablation at step 200 on 29 problems with 16 continuations per condition. Panel B uses 184 successful-trace boundary instances with 64 continuations. Panel C gives solver-constructed controls. The successful natural-language and successful-trace conditions include the source reasoning; the state-only conditions specify the listed arithmetic information.

Condition	Ckpt.	Problems × cont.	Prefix tokens	Rate
Panel A: state-only ablation				
Numeric multiset only	200	29 × 16	12.9	0.002
Multiset and first operator (bare label)	200	29 × 16	14.9	0.017
Successful natural-language scaffold	200	29 × 16	171.1	0.998
Panel B: successful-trace boundary contrast				
Before op_1	50	184 × 64	—	0.145
After op_1	50	184 × 64	—	0.914
Before op_1	275	184 × 64	—	0.617
After op_1	275	184 × 64	—	0.999
Panel C: solver-constructed controls				
Solver entrance after op_1, low-access family	275	solver set	—	0.133
Solver entrance after op_1, dominant family	275	solver set	—	0.193
Random syntactic after op_1 (any-valid)	275	solver set	—	0.082
Dead-end full expression (valid / format-closing)	275	20 × 8	—	0.000 / 1.000

E REPLICATION ON A PUBLIC GRPO RUN

Released checkpoints and protocol. The second Countdown setting uses the public checkpoint series `philschmid/qwen-2.5-3b-r1-countdown`, based on Qwen2.5-3B-Instruct and trained with TRL GRPO on `Jiayi-Pan/Countdown-Tasks-3to4`. We evaluate the released checkpoints in their native prompt and output format, including the assistant prefill that begins generation inside `<think>`.

Overlap-excluded evaluation set. The public data preparation shuffles the source dataset with seed 42 and takes the first 50,000 examples before its train and test split. Since the exact split cannot be reconstructed from the released code, we treat all 50,000 as a training superset, define the semantic key of a problem as its sorted input numbers with the target, and remove every match from our 150-problem set. Fifteen problems overlap, leaving 135 for the free-generation endpoint analyses.

Checkpoint selection. Checkpoint selection uses an independent 300-problem validation set disjoint from the 135-problem evaluation set by semantic key. We evaluate all eight released checkpoints with 64 samples per problem at temperature 0.7, top-$p = 0.9$, and a 1,024-token cap, and select the earliest checkpoint whose native-format rate exceeds 0.90. This rule selects step 25; step 450 is the late endpoint. Coverage declines steadily along the validation trajectory (Table 18), with a problem-level regression on $\log(\mathrm{step})$ giving slope -0.117 (95% CI $[-0.134, -0.099]$).

Table 18: Independent validation trajectory for the public GRPO run (300 problems, 64 samples per problem). The evaluation set used for endpoint comparisons is not used for checkpoint selection.

Checkpoint	pass@1	pass@64	Native-format	Solution cov.	Entrance cov.
Step 25	0.142	0.703	0.905	0.358	0.766
Step 50	0.181	0.710	0.947	0.384	0.735
Step 100	0.321	0.700	0.903	0.385	0.637
Step 150	0.450	0.657	0.864	0.304	0.540
Step 200	0.489	0.670	0.941	0.279	0.479
Step 300	0.507	0.673	0.947	0.267	0.479
Step 400	0.515	0.660	0.956	0.254	0.471
Step 450	0.516	0.663	0.955	0.258	0.473

E.1 Breadth at the selected endpoints

At 320 samples per problem, single-sample accuracy rises sharply while solution coverage, entrance coverage, and entrance entropy all narrow (Table 19). The supplemental reasoning-derived partition sharpens further, with think-entrance entropy falling from 2.225 to 0.081. We report the answer-defined partition as primary because it aligns with the solver-enumerated solution families used throughout.

Table 19: Endpoint evaluation for the public GRPO run (135 problems, 320 samples per problem).

Checkpoint	pass@1	pass@64	pass@256	Native-format rate	Solution cov.	Entrance cov.	Entrance H
Step 25	0.121	0.633	0.750	0.900	0.467	0.932	1.585
Step 450	0.429	0.570	0.621	0.941	0.268	0.571	0.760

E.2 Teacher-forced likelihood

Scoring the same 232 solver-constructed valid continuations at both selected checkpoints and splitting NLL at the first arithmetic operator, the increase is far larger at the entrance than during execution (Table 20). The per-token entrance shift is about eleven times the execution shift, reproducing the asymmetry under a different RLVR recipe and prompt format.

Table 20: Teacher-forced likelihood on the public GRPO run. Values are NLL per token with 95% problem-cluster bootstrap intervals.

Checkpoint or contrast	Entrance	Execution
Step 25	$6.840\ [6.741, 6.943]$	$1.078\ [0.995, 1.162]$
Step 450	$14.917\ [14.731, 15.119]$	$1.788\ [1.593, 1.984]$
Late − early	$+8.077\ [7.962, 8.193]$	$+0.709\ [0.578, 0.846]$

E.3 Supplied entrances

We repeat the entrance progression in the released model's native prompt, with 16 continuations per family and condition. The paired set contains 169 families on 111 problems whose supplied operand–operator cue matches the designated family at both checkpoints; complete calculations additionally match the solver witness's first exact reduction. At step 450, a minimal entrance raises designated-family completion from 0.126 to 0.227, and the completed first calculation reaches 0.265. Minimal-entrance completion rises from 0.160 to 0.227 across training.

Table 21 reports both unconditional outcomes and completion within native-format-valid outputs. Conditional rates are computed within each problem and then averaged; all 111 problems have format-valid outputs in every condition. Within these outputs, late-checkpoint designated completion rises from 0.126 to 0.237 after a minimal entrance.

Table 21: Supplied entrances on public GRPO: 111 paired problems, 169 matched families, and 16 continuations per family and condition. The first two outcome columns retain every generated continuation in the denominator. The last column conditions on native-format-valid outputs within each problem.

Checkpoint	Prefix	Designated	Any-valid	Native format	Designated given format
Step 25	None	0.065	0.135	0.906	0.071
Step 25	Minimal	0.160	0.249	0.752	0.193
Step 25	First calculation	0.171	0.233	0.851	0.190
Step 450	None	0.126	0.494	0.952	0.126
Step 450	Minimal	0.227	0.416	0.816	0.237
Step 450	First calculation	0.265	0.391	0.824	0.276

Table 22: Paired intervention outcomes on 100 problems, 64 samples per problem. Differences are relative to the same step-275 control, with 95% problem-bootstrap intervals (10,000 draws, seed 1729). † marks pass@1 equivalence under the ± 0.02 TOST margin.

Intervention	pass@1	Δ pass@1	Solution cov.@64	Δ coverage
Step-275 policy (control)	0.278	—	0.111	—
Prompt diversification	$0.282^{\dagger}$	$+0.004$ $[-0.008, 0.017]$	0.125	$+0.014$ $[0.000, 0.038]$
Forced alternative operator	0.286	$+0.008$ $[-0.057, 0.074]$	0.104	-0.007 $[-0.016, 0.002]$
Answer-only logit mixing	$0.269^{\dagger}$	-0.009 $[-0.016, -0.003]$	0.105	-0.006 $[-0.014, 0.000]$
High temperature (T=1.0)	0.264	-0.014 $[-0.026, -0.004]$	0.127	$+0.016$ $[0.007, 0.027]$
Reasoning-phase logit mixing	$0.277^{\dagger}$	-0.002 $[-0.020, 0.017]$	0.124	$+0.013$ $[0.004, 0.025]$
Layer interpolation	0.305	$+0.026$ $[-0.009, 0.064]$	0.152	$+0.041$ $[0.020, 0.068]$
Checkpoint sampling	0.161	-0.117 $[-0.155, -0.080]$	0.211	$+0.100$ $[0.059, 0.146]$

F Reopening Entrances (PPO)

The intervention ledger fixes 100 comparison problems within the 150-problem PPO evaluation pool. Every arm uses 64 samples per problem, matched problem identities, and the same step-275 control. We use 10,000 paired problem-bootstrap draws (seed 1729); pass@1 equivalence is assessed by paired two one-sided tests with margin ± 0.02. Table 4 gives point estimates and Table 22 gives paired intervals. The historical layer sweep is reported in Appendix F.6.

Intervention definitions. Prompt diversification adds a request for a different valid method. The forced operator arm constrains the opening operator without supplying an arithmetic state (Appendix F.1). Answer-only and reasoning-phase logit mixing interpolate step-50 logits either inside the answer span or during reasoning, respectively. High-temperature decoding raises the sampling temperature from 0.7 to 1.0. Layer interpolation replaces blocks 20–28 by the midpoint of their step-50 and step-275 parameters. Checkpoint sampling allocates 32 of the 64 rollouts to each endpoint checkpoint.

Recovery pattern. Prompt diversification and reasoning-phase mixing yield similar solution-coverage gains ($+0.014$ and $+0.013$). Layer interpolation yields the largest gain among the single-policy conditions, $+0.041$ $[0.020, 0.068]$, with observed pass@1 of 0.305 versus 0.278. Checkpoint sampling yields $+0.100$ $[0.059, 0.146]$ in coverage while moving single-sample accuracy toward the early checkpoint.

Table 23: Public GRPO intervention comparison on 100 problems, 64 samples per problem. Differences use the common step-450 control and 95% paired problem-bootstrap intervals.

Intervention	pass@1	Δ pass@1	Solution cov.@64	Δ coverage	Family cov.@64
Step-450 control	0.407	—	0.169	—	0.285
Answer-only mixing	0.407	-0.001 $[-0.011, 0.008]$	0.182	$+0.013$ $[-0.008, 0.040]$	0.314
Reasoning-phase mixing	0.390	-0.018 $[-0.041, 0.003]$	0.176	$+0.007$ $[-0.021, 0.037]$	0.301
Layer interpolation	0.379	-0.028 $[-0.048, -0.010]$	0.172	$+0.003$ $[-0.027, 0.028]$	0.297
Checkpoint sampling	0.247	-0.160 $[-0.201, -0.120]$	0.268	$+0.099$ $[0.060, 0.141]$	0.416

F.1 Forced operator override

Starting from successful step-275 samples, the forced-operator arm identifies an alternative first operator that can open at least one valid expression under the same first number, without supplying an arithmetic state. On the confirmation split it changes solution coverage by -0.007 $[-0.016, 0.002]$. This intervention varies the operator while keeping the model's first operand fixed; the supplied entrance in §5 specifies both.

F.2 Cross-run confirmation on public GRPO

We evaluate the same intervention families on the independent public GRPO trajectory. Answer-only mixing gives a pass@1 difference of -0.0006 $[-0.0114, 0.0084]$, within the ± 0.02 equivalence margin, and increases canonical entrance-family coverage by $+0.030$ $[0.001, 0.066]$. Checkpoint sampling produces the largest solution-coverage gain, $+0.099$, and increases family coverage by $+0.132$. With the same layer-interpolation setting, pass@1 is 0.379 versus 0.407 for the late policy, and solution coverage is 0.172 versus 0.169. Table 23 gives all paired outcomes.

F.3 Entrance allocation

For each problem we take up to three failed samples as scaffolds, append each feasible solver entrance, and sample 16 continuations per entrance. Budget matching uses one per-problem token cap, the minimum of the full 320-sample generated-token budget and the full allocation cost, and both conditions are truncated deterministically to it.

At step 275, 128 problems have complete outputs in both conditions. There, entrance allocation exceeds same-problem free resampling by $+0.148$ $[0.086, 0.211]$ in any-valid success, $+0.077$ $[0.048, 0.109]$ in coverage, and $+0.430$ $[0.297, 0.570]$ in distinct entrance families. At step 50 all 150 problems are comparable and the contrasts are flat: any-valid -0.040 $[-0.107, 0.027]$ and coverage -0.001 $[-0.044, 0.042]$. Allocating attempts across feasible entrances therefore helps precisely once access has concentrated.

F.4 Entrance recovery under interpolation

Canonical-family coverage counts distinct families among valid canonical solutions, normalized by the solver's family count. Generated entrance entropy is Shannon entropy over the first parsed operand–operator pair in each reasoning block, including incorrect outputs. It is calculated within each problem over detected entrances and then averaged. Table 24 reports both quantities and detection rates. For layer interpolation, the paired changes are $+0.067$ $[0.034, 0.104]$ in canonical-family coverage and $+0.137$ $[0.105, 0.170]$ nats in generated entrance entropy.

The historical unobserved set is fixed from Appendix D.6: access at least 0.05 at step 50 and no hit in 320 step-275 samples. Its intersection with these 100 problems contains 18 families on 17 problems. Layer interpolation revisits 7 families and successfully completes 6; the fresh step-275 control successfully revisits none. The problem-macro successful-revisit rate rises from 0 to 0.353 $[0.118, 0.588]$. Table 25 retains all intervention conditions.

Table 24: Entrance outcomes on the 100-problem paired comparison, 64 rollouts per problem. Family coverage uses valid canonical solutions; generated-entrance entropy H includes detected entrances in incorrect outputs. Differences are against the same late control, with 95% paired problem-bootstrap intervals.

Intervention	Family cov.@64	Δ family cov.	Generated H	ΔH	Detection
Step-275 control	0.185	—	0.046	—	1.0000
Prompt diversification	0.201	$+0.017$ $[0.000, 0.040]$	0.062	$+0.016$ $[0.003, 0.030]$	1.0000
Forced alternative operator	0.174	-0.010 $[-0.030, 0.009]$	0.117	$+0.071$ $[0.026, 0.122]$	1.0000
Answer-only mixing	0.169	-0.015 $[-0.040, 0.000]$	0.043	-0.003 $[-0.011, 0.004]$	1.0000
High temperature	0.212	$+0.028$ $[0.012, 0.046]$	0.113	$+0.067$ $[0.047, 0.089]$	0.9956
Reasoning-phase mixing	0.209	$+0.025$ $[0.009, 0.045]$	0.093	$+0.047$ $[0.026, 0.071]$	1.0000
Layer interpolation	0.251	$+0.067$ $[0.034, 0.104]$	0.183	$+0.137$ $[0.105, 0.170]$	0.9998
Checkpoint sampling	0.315	$+0.130$ $[0.081, 0.185]$	0.703	$+0.657$ $[0.619, 0.694]$	0.9295

Table 25: Revisits to the fixed historical unobserved set: 18 families on 17 comparison problems. Counts sum distinct families; the final column averages each problem's fraction of successfully revisited historical families and gives a 95% problem-bootstrap interval.

Intervention	Revisited	Successful	Problem-macro successful revisit
Step-275 control	0/18	0/18	0.000
Prompt diversification	2/18	2/18	0.118 [0.000, 0.294]
Forced alternative operator	1/18	1/18	0.029 [0.000, 0.088]
Answer-only mixing	0/18	0/18	0.000 [0.000, 0.000]
High temperature	2/18	2/18	0.118 [0.000, 0.294]
Reasoning-phase mixing	1/18	0/18	0.000 [0.000, 0.000]
Layer interpolation	7/18	6/18	0.353 [0.118, 0.588]
Checkpoint sampling	16/18	13/18	0.706 [0.471, 0.882]

F.5 Intermediate-checkpoint comparison

We compare step 150 with the late policy and interpolation on the same 100 problem keys (Table 26). The archival step-150 pool contains 320 samples per problem. For a solution observed c times, its probability of appearing in a uniformly drawn 64-sample subset is

$$1 - \frac{\binom{320-c}{64}}{\binom{320}{64}}.$$

Summing this probability over solutions or families gives the expected coverage at the matched budget. Interpolation has coverage 0.152 versus 0.141 for step 150, a paired difference of $+0.011$ $[-0.007, 0.037]$, with pass@1 of 0.305 versus 0.334. Both recover breadth relative to step 275.

Table 26: Intermediate-checkpoint comparison on the same 100 problem keys. The step-150 row is the exact expectation over uniform 64-sample subsets of its 320-sample pool; the other rows use their fixed 64-sample runs.

Policy	pass@1	pass@64	Solution cov.@64	Family cov.@64
Step 150	0.334	0.436	0.141	0.242
Step 275	0.278	0.370	0.111	0.185
Layer interpolation	0.305	0.450	0.152	0.251

Table 27: Historical layer sweep at $\alpha = 0.50$ (150 problems, 64 samples per problem), with pure-checkpoint controls. Global mixing includes the full parameter set; local bands mix the listed transformer blocks. All 56 configurations are provided in `layer_all_56_configs.csv`.

Interpolated parameters	pass@1	pass@64	Solution cov.@64
Step 50	0.053	0.493	0.226
Step 275	0.286	0.373	0.099
Global weight mixture	0.256	0.487	0.182
$[0, 12)$	0.285	0.380	0.105
$[12, 24)$	0.262	0.420	0.132
$[16, 24)$	0.269	0.420	0.124
$[20, 28)$	0.285	0.453	0.148
$[24, 32)$	0.309	0.440	0.135
$[28, 36)$	0.297	0.387	0.112
$[32, 36)$	0.288	0.393	0.108
$[0, 36)$	0.256	0.487	0.186

Table 28: Position-controlled logit mixing on 150 problems, 64 samples each, with $\alpha = 0.70$. Entrance metrics use generated reasoning entrances. The two fixed windows both mix exactly eight tokens.

Mixing region	pass@1	Solution cov.	Entrance cov.	Entrance H	Mixed tokens
Step-275 control	0.284	0.098	0.161	0.030	0.0
First 8 reasoning tokens	0.268	0.114	0.202	0.108	8.0
8 tokens after entrance	0.278	0.105	0.179	0.035	8.0
Through entrance	0.264	0.125	0.224	0.158	62.1
After entrance	0.269	0.140	0.170	0.034	166.3
Full reasoning	0.250	0.142	0.207	0.152	233.2

F.6 Layer and mixing-weight sweep

The historical sweep evaluates 56 configurations on 150 problems with 64 samples each. Global mixtures use six early-checkpoint weights, $\alpha \in \{0.10, 0.20, 0.30, 0.50, 0.70, 1.00\}$, alongside local layer-band mixtures and pure-checkpoint controls. Table 27 shows the fixed-$\alpha = 0.50$ layer comparison; the complete grid is included in the supplementary CSV. A band $[a, b)$ replaces transformer blocks a through $b - 1$ with $(1 - \alpha)\theta_{275} + \alpha\theta_{50}$. The main paired comparison uses $[20, 28)$ and $\alpha = 0.50$.

F.7 Equal-length position interventions

A separate 150-problem experiment compares step-50 logit mixing at different reasoning positions, with 64 samples per problem and mixing weight $\alpha = 0.70$. The fixed-window conditions mix eight tokens at the start of reasoning or immediately after the detected entrance. Table 28 includes both windows, the late control, and longer mixing conditions. Relative to the post-entrance window, the early window raises entrance coverage by $+0.023$ $[0.006, 0.046]$ and entropy by $+0.073$ $[0.054, 0.094]$ nats; the solution-coverage difference is $+0.009$ $[-0.001, 0.020]$ and the pass@1 difference is -0.010 $[-0.025, 0.005]$.

G Math Benchmarks: Early Concentration and Its Limits

Exhaustive enumeration is unavailable here, so the transfer test uses observable structure: diversity of first calculations, distinct traces, and same-trace likelihood shifts around the first complete calculation.

Table 29: Origin-stratified difference-in-differences for Qwen2.5-7B, in nats per token. Intervals use 10,000 problem-bootstrap draws (seed 0). Counts give the problems and traces entering the paired-origin estimate.

Benchmark	DiD	95% CI	Problems	Traces
AIME24	0.228	$[0.154, 0.318]$	9	112
AMC23	0.307	$[0.264, 0.352]$	36	482
GSM8K	0.313	$[0.302, 0.325]$	491	7728
MATH500	0.337	$[0.320, 0.355]$	460	5993
Minerva	0.193	$[0.171, 0.216]$	132	1349
OlympiadBench	0.273	$[0.254, 0.292]$	322	3958

Table 30: Components of same-trace scoring: Base-minus-SimpleRL log-likelihood in nats per token, averaged by problem within each trace origin. Base-origin rows supply Figure 3(a). Each origin uses its available trace pool; Table 29 pairs origins on shared problem keys. Intervals use 10,000 problem-bootstrap draws.

Benchmark	Trace origin	Early segment	Execution	Problems	Traces
GSM8K	Base	$0.303\ [0.292, 0.314]$	$0.081\ [0.077, 0.085]$	494	5198
GSM8K	RLVR	$-0.136\ [-0.139, -0.134]$	$-0.046\ [-0.048, -0.045]$	491	2538
MATH500	Base	$0.344\ [0.327, 0.361]$	$0.089\ [0.084, 0.095]$	466	3605
MATH500	RLVR	$-0.126\ [-0.130, -0.122]$	$-0.045\ [-0.047, -0.043]$	467	2413
Minerva	Base	$0.176\ [0.155, 0.199]$	$0.056\ [0.048, 0.066]$	149	742
Minerva	RLVR	$-0.135\ [-0.142, -0.129]$	$-0.047\ [-0.050, -0.044]$	159	757
OlympiadBench	Base	$0.243\ [0.225, 0.262]$	$0.052\ [0.048, 0.055]$	348	2233
OlympiadBench	RLVR	$-0.127\ [-0.131, -0.122]$	$-0.045\ [-0.047, -0.043]$	350	1842
AMC23	Base	$0.267\ [0.223, 0.313]$	$0.056\ [0.046, 0.066]$	37	236
AMC23	RLVR	$-0.127\ [-0.138, -0.117]$	$-0.038\ [-0.043, -0.033]$	37	248
AIME24	Base	$0.213\ [0.141, 0.292]$	$0.040\ [0.030, 0.053]$	13	58
AIME24	RLVR	$-0.124\ [-0.145, -0.104]$	$-0.045\ [-0.056, -0.036]$	13	71

G.1 SAME-TRACE SCORING

For trace origin $o \in \{\text{base}, \text{RLVR}\}$ and benchmark d, let $C_{o,d}$ and $E_{o,d}$ be base minus RLVR per-token log-likelihood differences in the early and execution segments. We report

$$\text{DiD}_d = (C_{\text{base},d} - E_{\text{base},d}) - (C_{\text{RLVR},d} - E_{\text{RLVR},d}). \tag{7}$$

On Qwen2.5-7B every estimate is positive and all six intervals lie above zero (Table 29). The origin-stratified contrast localizes the between-policy change before the first complete calculation. We average traces within each problem and origin, pair the two origins by problem, and bootstrap the paired problem contrasts. Table 30 gives the early and execution components.

Same-source style normalization. Minimal normalization converts curly quotes to straight quotes, collapses runs of spaces or tabs, reduces three or more consecutive newlines to two, and trims the outer whitespace. We pair raw and normalized records by benchmark, problem, generating model, sample index, and original-text hash. For each benchmark we retain the same source trajectories in both styles and pair their Base- and RLVR-origin problem means. Table 31 reports the resulting DiD estimates and effective sample counts used in Figure 3(b). All six normalized estimates remain positive; on GSM8K the value decreases from 0.265 to 0.184 nats per token.

G.2 SEMANTIC BOUNDARY RESEGMENTATION

The first-calculation boundary in §7 is extracted by a deterministic rule parser. We independently resegment the same trace pool using blinded model-based annotations of the earliest complete numeric calculation: the annotator receives only the prompt, response, and numbered text units, without model identity, trace origin, or parser output, and a deterministic resolver maps the returned span to token offsets. The resolver succeeds on 19,160 of the 19,941 scored traces (96.1%), and failure rates differ by at most 1.85 percentage points between Base- and RLVR-origin traces across benchmarks.

Table 31: Style normalization on identical source trajectories. Both versions use the same paired-origin problems and traces. Values are origin-stratified DiD in nats per token, with 95% problem-bootstrap intervals (10,000 draws, seed 0); these values supply Figure 3(b).

Benchmark	Raw, same traces	Style-normalized	Problems	Traces
GSM8K	$0.265\ [0.249, 0.281]$	$0.184\ [0.168, 0.200]$	335	2831
MATH500	$0.292\ [0.265, 0.318]$	$0.280\ [0.251, 0.309]$	219	1824
Minerva	$0.169\ [0.134, 0.207]$	$0.160\ [0.115, 0.210]$	40	288
OlympiadBench	$0.279\ [0.251, 0.310]$	$0.278\ [0.248, 0.310]$	173	1350
AMC23	$0.279\ [0.211, 0.356]$	$0.306\ [0.234, 0.384]$	21	192
AIME24	$0.172\ [0.126, 0.214]$	$0.150\ [0.087, 0.213]$	4	50

Using the same per-token NLL caches, we split each resolved trace immediately before its annotated first calculation. For each trace origin, we report the Base-minus-SimpleRL per-token log-likelihood difference in the early segment minus that in the execution segment, negating the corresponding NLL shift (Table 32). Under this log-likelihood convention, Base-origin shifts are positive and RLVR-origin shifts are negative on all six benchmarks, with all problem-cluster intervals excluding zero. The localization is therefore not an artifact of the rule-based boundary parser. Canonicalizing the annotated spans likewise preserves the corpus-level diversity gap, with first-calculation vocabulary entropy 7.438 versus 7.050 on GSM8K and 7.412 versus 7.000 on MATH500 (Base versus SimpleRL).

Table 32: Semantic-boundary early-minus-execution log-likelihood shift for Qwen2.5-7B Base versus SimpleRL, stratified by the policy that generated the scored trace. Intervals use 10,000 problem-cluster bootstrap draws.

Benchmark	Base-origin shift	RLVR-origin shift	Base traces	RLVR traces
AIME24	$+0.083\ [0.056, 0.111]$	-0.055 $[-0.072, -0.039]$	58	71
AMC23	$+0.142\ [0.109, 0.178]$	-0.060 $[-0.069, -0.050]$	230	242
GSM8K	$+0.226\ [0.212, 0.240]$	-0.114 $[-0.117, -0.110]$	5133	2500
MATH500	$+0.180\ [0.166, 0.194]$	-0.064 $[-0.068, -0.060]$	3404	2306
Minerva	$+0.079\ [0.064, 0.096]$	-0.055 $[-0.061, -0.050]$	689	690
OlympiadBench	$+0.113\ [0.103, 0.123]$	-0.062 $[-0.066, -0.058]$	2077	1760

A local linear regression in a ± 20-token window around the semantic boundary attributes the shift to a broad early phase rather than a single-token jump, so we use the first calculation as a semantic alignment point, not as a claimed discontinuity.

G.3 Sampling depth at 256 samples

Re-evaluating the Qwen2.5-7B pair on GSM8K and MATH500 with 256 samples over 200 problems per benchmark, large-budget accuracy nears saturation for both policies while first-calculation and trace diversity stay several-fold lower under SimpleRL (Table 33).

Table 33: Sampling depth at 256 samples per problem, 200 problems per benchmark.

Model	Benchmark	pass@64	pass@256	Distinct first calcs.	First-calc. H	Distinct traces
Base 7B	GSM8K	0.993	1.000	8.98	1.054	148.5
SimpleRL 7B	GSM8K	0.973	0.975	2.20	0.247	22.0
Base 7B	MATH500	0.950	0.965	10.20	0.829	158.9
SimpleRL 7B	MATH500	0.937	0.950	3.32	0.288	25.2

G.4 Conditional recovery from supplied arithmetic states

For standard math tasks we construct an analogue of an entrance family from calculations that the base policy reaches in free sampling. From 64 base rollouts per problem we retain up to the two most frequent first-calculation families and measure the RLVR policy after supplying that completed calculation. Table 34 separates families that the RLVR policy does and does not reach in its own 64-sample free rollouts.

Table 34: Conditional execution from base-reachable first-calculation families. Intervals use problem-cluster bootstrap.

Benchmark	Families	RLVR-unreached	Overall completion	Unreached completion
GSM8K	384	224	0.863 [0.823, 0.901]	0.850 [0.801, 0.898]
MATH500	337	155	0.715 [0.661, 0.768]	0.713 [0.645, 0.780]

First-calculation information levels. On 371 four-number Countdown problems with solver-consistent supplied states, we compare steps 50 and 275 with 64 continuations per state. Information level $d = 0$ supplies no entrance, $d = 1$ gives the first operand, $d = 2$ gives the first binary operation, and $d = 3$ adds its exact result. Table 35 reports the four levels. For each problem i, let $\Delta_i(d)$ be late minus early completion; we fit

$$\Delta_i(d) = a_i + \gamma_i d + \epsilon_{i,d}, \qquad \gamma = \frac{1}{371}\sum_i \gamma_i.$$

The mean slope is $\gamma = -0.02194$ with 95% problem-bootstrap interval $[-0.03213, -0.01210]$ (10,000 draws). The late–early completion difference decreases as more information about the first calculation is supplied.

H Scale, Pipelines, and Supervised Controls

The two Countdown runs identify the same regime under PPO and GRPO. This appendix asks whether early concentration recurs at larger scale and whether higher accuracy requires the same loss of first-calculation breadth.

H.1 Scale comparison

The 7B and 14B Qwen2.5 base and SimpleRL pairs move in the same direction: at both scales RLVR raises single-sample accuracy while first-calculation entropy falls sharply, and the matched GSM8K correct-only comparison agrees after conditioning on correctness (Table 36). First-calculation entropy is the primary comparison because longer traces do not inflate it; the distinct-trace rate in Table 5 is its length-sensitive complement.

H.2 RL after distillation

We compare `DeepSeek-R1-Distill-Qwen-1.5B` with its RL-trained successor `DeepScaleR-1.5B-Preview`, released by DeepSeek and Agentica, respectively. The

Table 35: First-calculation information levels on 371 paired four-number problems. Every state receives 64 continuations per checkpoint. Values are any-valid completion rates with 95% problem-bootstrap intervals.

Level	Supplied information	Step 50	Step 275
0	No entrance	0.017 [0.014, 0.019]	0.160 [0.127, 0.194]
1	First operand	0.010 [0.008, 0.012]	0.069 [0.057, 0.082]
2	First binary operation	0.072 [0.063, 0.081]	0.139 [0.120, 0.158]
3	First operation and result	0.077 [0.068, 0.087]	0.145 [0.125, 0.166]

Table 36: Scale comparison for the Qwen base and SimpleRL pair. Macro columns average six benchmarks; the last column is matched GSM8K correct-only first-calculation entropy.

Model	pass@1	pass@64	First-calc. H	GSM8K correct-only H
Qwen Base 7B	0.379	0.754	1.063	0.894
Qwen SimpleRL 7B	0.510	0.760	0.744	0.285
Qwen Base 14B	0.377	0.774	1.207	0.970
Qwen SimpleRL 14B	0.546	0.774	0.711	0.300

latter uses GRPO with verifiable mathematical-answer rewards; we evaluate the released checkpoints using the benchmark protocol in Appendix B.[3] RL applied to the distilled policy gives a different profile: single-sample accuracy rises while large-budget accuracy and first-calculation structure stay level, and the small entropy changes point in opposite directions on GSM8K and MATH500 (Table 37). With the OLMo-3 comparison in Table 5, accuracy gains arrive with distinctly different breadth profiles.

Table 37: RL after distillation at 1.5B; first-calculation statistics are measured at the first complete calculation, 1,000 bootstrap draws.

Model	Benchmark	pass@1	pass@64	First-calc. H	Effective first calcs.
Distilled base 1.5B	GSM8K	0.701	0.985	1.126	3.67
RL over distill 1.5B	GSM8K	0.766	0.980	1.206	4.02
Distilled base 1.5B	MATH500	0.633	0.955	1.168	4.26
RL over distill 1.5B	MATH500	0.732	0.955	1.130	3.93

H.3 Supervised fine-tuning controls on Countdown

To assess whether solution narrowing is an inevitable consequence of learning correct reasoning paths, we train supervised fine-tuning (SFT) baselines starting from Qwen2.5-3B (Base).

Disjoint multi-solution supervision. We construct 8,000 training problems (16,000 in the data-size ablation) and 500 validation problems disjoint from the 135 held-out evaluation tasks by semantic key (sorted numbers + target). For each training problem, a solver enumerates all valid canonical expressions. We extract up to k solutions per problem using an entrance-diverse round-robin strategy across opening families and render them into native Countdown reasoning traces using 12 templated prompt variations. All generated traces undergo automated mathematical verification, canonical parser validation, and deduplication. The *uniform* condition samples up to k canonical solutions per problem uniformly without replacement, using the same data and rendering pipeline.

Validation-based checkpoint selection. Models are fine-tuned using LoRA (Hu et al., 2022) (rank 32, $\alpha = 64$, learning rate 10^{-4}, 2 epochs, batch size 16), with checkpoints saved every 250 optimizer steps. Checkpoint selection is performed on the 500 independent validation tasks using the following rule

$$\hat{c} = \arg \min_{c:\ \mathrm{format}(c) \geq 0.90} \left| \widehat{\mathrm{pass@1}}_{\mathrm{val}}(c) - 0.429 \right|, \tag{8}$$

which selects the checkpoint closest to the GRPO step-450 operating point (0.429) subject to native-format validity above 0.90. The 135 evaluation problems are then evaluated with 320 samples only at the selected checkpoint; Table 38 reports the resulting SFT operating points.

[3] Model sources: `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B` and `https://huggingface.co/agentica-org/DeepScaleR-1.5B-Preview`.

Table 38: Dose–response ablation over supervised solutions per problem k on 135 held-out Countdown tasks (320 samples per problem). All intervals are 2,000 problem-bootstrap draws (seed 1729).

Supervision condition	pass@1	pass@64	pass@256	Solution cov.	Entrance cov.
GRPO step 450 (RLVR)	0.429	0.570	0.621	0.268	0.571
SFT (k=1)	0.120 $[0.095, 0.145]$	0.737	0.856	0.471 $[0.412, 0.532]$	0.604
SFT (k=2)	0.221 $[0.176, 0.266]$	0.806	0.893	0.551 $[0.494, 0.609]$	0.766
SFT (k=4, 8k)	0.310 $[0.254, 0.366]$	0.853	0.923	0.725 $[0.669, 0.782]$	0.825
SFT (k=4, 16k)	0.333 $[0.274, 0.389]$	0.855	0.916	0.709 $[0.650, 0.764]$	0.855
SFT (k=4, uniform)	0.269 $[0.219, 0.315]$	0.836	0.928	0.779 $[0.725, 0.835]$	0.821
SFT (k=8)	0.293 $[0.241, 0.344]$	0.836	0.903	0.766 $[0.710, 0.822]$	0.818

Dose–response relationship. As shown in Table 38, solution coverage exhibits a strict monotonic increase with supervision breadth k at 8,000 training problems ($0.471 \rightarrow 0.551 \rightarrow 0.725 \rightarrow 0.766$). Even with a single supervised path per problem (k=1), SFT preserves 0.471 coverage, exceeding the late RL policy (0.268). This comparative demonstration indicates that solution space contraction is not an unavoidable byproduct of learning correct reasoning paths, but reflects the selective reinforcement dynamics of on-policy RL.

Continuation under GRPO. Starting from the k=4, 8,000-problem SFT checkpoint (step 2,750; coverage 0.725), we continue on-policy GRPO for 500 training steps under the native Countdown reward. The run uses learning rate 10^{-6}, two rollouts per prompt, and a KL coefficient of 0.001. Across the last 100 logged training steps, mean rollout response entropy is 0.084 nats. Training-response probabilities therefore concentrate again from the broad supervised initialization.

I Raw Trajectories for Qualitative Case Studies

This section gives the full text of selected completions at $T = 0.6$, top-$p = 0.95$ for the case studies in §7. The supplementary data contain all 960 rollouts (three problems, five models, and 64 samples per model), with original sample identifiers. Line wrapping is added for typesetting.

I.1 GSM8K Item 108 (Baking Cookies, Gold Answer: 50)

Question. Henry wants to bake twice as many cookies as last year. When he finishes baking, he realizes he actually baked 15 more cookies than he meant to. He drops 5 of his cookies as he is putting them out to cool, and now has a total of 110 cookies. How many cookies did Henry bake last year?

Qwen Base 7B — Reverse execution route (Sample 6; correct).

```
To find out how many cookies Henry baked last year, we can follow these
steps:

1. **Determine the number of cookies Henry meant to bake this year:**
   - Henry now has 110 cookies.
   - He dropped 5 cookies, so before dropping them, he had \(110 + 5 =
   115\) cookies.
   - He baked 15 more cookies than he meant to, so the number of cookies
   he meant to bake is \(115 - 15 = 100\) cookies.

2. **Calculate the number of cookies Henry baked last year:**
   - Henry wants to make twice as many cookies as he did last year.
```

```
   - This year, he meant to bake 100 cookies, which is twice the number
   he baked last year.
   - Therefore, the number of cookies he baked last year is \(100 \div 2
   = 50\) cookies.

So, the number of cookies Henry baked last year is \(\boxed{50}\).
```

Qwen SimpleRL 7B — Forward equation (Sample 0; correct).

```
To determine how many cookies Henry baked last year, we will follow these
steps:

1. **Define the variable:**
   Let \( x \) be the number of cookies Henry baked last year.

2. **Calculate the number of cookies Henry intended to bake this year:**
   Since Henry wants to make twice as many cookies as he did last year,
   he intended to bake \( 2x \) cookies this year.

3. **Account for the extra cookies baked:**
   Henry actually baked 15 more cookies than he intended, so he baked \(
   2x + 15 \) cookies.

4. **Account for the cookies dropped:**
   Henry dropped 5 cookies, so the number of cookies he has now is \( (2x
   + 15) - 5 \).

5. **Set up the equation:**
   According to the problem, Henry now has a total of 110 cookies.
   Therefore, we can set up the following equation:
   \[
   (2x + 15) - 5 = 110
   \]

6. **Simplify the equation:**
   \[
   2x + 10 = 110
   \]

7. **Solve for \( x \):**
   \[
   2x = 110 - 10
   \]
   \[
   2x = 100
   \]
   \[
   x = \frac{100}{2}
   \]
   \[
   x = 50
   \]

So, Henry baked \(\boxed{50}\) cookies last year.
```

OLMo-3 7B RLVR — Reverse execution route (Sample 12; correct).

```
Let's define variables and break the problem into steps.

Let \( x \) be the number of cookies Henry baked last year.

Henry wants to make **twice as many cookies as last year** this year. So
his intended amount this year is:
\[
2x
\]
```

```
But the problem says:
*"When he finishes baking, he realizes he actually baked 15 more cookies
than he meant to."*
So the actual number he baked this year is:
\[
2x + 15
\]

Next, Henry drops 5 cookies while cooling, and then he has a total of 110
cookies left. So before dropping the 5 cookies, he had:
\[
110 + 5 = 115 \text{ cookies}
\]

This 115 cookies is the actual number he baked (since he drops the
cookies after baking):
\[
2x + 15 = 115
\]

Now solve for \( x \):

\[
2x + 15 = 115
\]
\[
2x = 115 - 15
\]
\[
2x = 100
\]
\[
x = \frac{100}{2}
\]
\[
x = 50
\]

So, Henry baked \(\boxed{50}\) cookies last year.
```

I.2 GSM8K ITEM 392 (STREAMING BUNDLE, GOLD ANSWER: 34)

Question. Tim decides to cancel his cable subscription and get streaming services. He gets Netflix for $10 a month. Hulu and Disney Plus normally cost $10 a month each but he saves 20% for bundling. How much money does he save by cancelling his $60 cable package?

Qwen SimpleRL 7B — Initial bundle sum (Sample 4; correct).

```
To determine how much money Tim saves by cancelling his $60 cable
package, we need to calculate the total cost of the streaming services
and then compare it to the cost of the cable package.

1. **Calculate the cost of Netflix:**
   Netflix costs $10 per month.

2. **Calculate the cost of Hulu and Disney Plus before the discount:**
   Both Hulu and Disney Plus normally cost $10 per month each. So, the
   total cost for both before the discount is:
   \[
   10 + 10 = 20 \text{ dollars per month}
   \]

3. **Calculate the discount for bundling Hulu and Disney Plus:**
   Tim gets a 20% discount for bundling. The discount amount is:
   \[
   20 \times 0.20 = 4 \text{ dollars}
   \]
```

```
   Therefore, the total cost for Hulu and Disney Plus after the discount
   is:
   \[
   20 - 4 = 16 \text{ dollars per month}
   \]

4. **Calculate the total cost of all streaming services:**
   Adding the cost of Netflix and the discounted cost of Hulu and Disney
   Plus gives:
   \[
   10 + 16 = 26 \text{ dollars per month}
   \]

5. **Calculate the savings by cancelling the cable package:**
   The cable package costs $60 per month. By switching to the streaming
   services, Tim pays $26 per month. The savings is:
   \[
   60 - 26 = 34 \text{ dollars per month}
   \]

Thus, the amount of money Tim saves by cancelling his $60 cable package
is \boxed{34}.
```

OLMo-3 7B RLVR — Per-item savings (Sample 3; correct).

```
Let's break the problem down step by step:

**1. Cost of the old cable subscription:**
Tim was paying $60 per month for cable.

---

**2. Cost of the new streaming services:**
He is switching to Netflix, Hulu, and Disney Plus.

- **Netflix:** $10 per month (no discount)
- **Hulu:** Normally $10 per month, but with a 20% discount for bundling.
- **Disney Plus:** Normally $10 per month, but also gets the 20% bundle
discount (assuming all three are bundled together for the discount).

Let's calculate the discounted Hulu and Disney Plus:

- 20% off $10 = $10 × 0.20 = $2 off
  So each of Hulu and Disney Plus will cost $10 - $2 = **$8 per month**
  (due to the bundle).

Therefore, total monthly cost for streaming:
- Netflix: $10
- Hulu: $8
- Disney Plus: $8

Total streaming cost per month = $10 + $8 + $8 = **$26**

---

**3. Savings calculation:**
Old cost: $60
New cost: $26
Savings per month = $60 - $26 = **$34**

---

**Final Answer:**
\boxed{34}
```

OLMo-3 7B RLVR — Discounted item price (Sample 49; correct).

```
Let's break down the problem step by step:
```

```
**1. Cost of the old cable subscription:**
Tim was paying $60 per month for cable.

---

**2. Cost of new streaming services:**

- **Netflix:** $10 per month (no discount).
- **Hulu:** Normally $10/month, but with a 20% discount for bundling.
- **Disney Plus:** Also normally $10/month, but the discount applies here
too (assuming bundling includes all three).

Let's calculate the bundled discount for Hulu and Disney Plus.

**Bundling Discount:**
- 20% off means he pays 80% of the original price.
- 80% of $10 = 0.8 × 10 = $8 per month per service.

So, for each of Hulu and Disney Plus, he pays $8 per month when bundled.

**Total monthly cost for streaming:**
- Netflix: $10
- Hulu: $8
- Disney Plus: $8

Total = $10 + $8 + $8 = **$26 per month**

---

**3. Money saved compared to cable:**

Old cost (cable): $60
New cost (streaming): $26

Money saved = $60 - $26 = **$34 per month**

---

**Final Answer:**

\boxed{34}
```

I.3 Companion Analysis on MATH500 (Item 451)

On MATH500 Item 451 (map scale ratio; gold answer $639/40$), all models achieve $pass@64 = 1$. Qwen Base accesses 10 distinct correct first-step forms ($H = 1.590$ nats, $pass@1 = 0.797$). Direct RLVR (Qwen SimpleRL) elevates $pass@1$ to 1.000 while collapsing the correct support to only 2 forms ($H = 0.525$ nats, with 50/64 rollouts entering via $15 + 0.975$). In contrast, the staged OLMo-3 progression preserves opening diversity ($7 \rightarrow 10 \rightarrow 13$ correct forms; $H : 1.678 \rightarrow 1.696 \rightarrow 1.804$ nats across SFT, DPO, and RLVR), confirming that multi-stage entropy preservation generalizes across distinct mathematical reasoning benchmarks.

Question. On a particular map, 3 inches on the map equates to 10 miles in real life. If you know that the real life distance between two buildings on the map is 53.25 miles, what would the distance between the buildings be (in inches) on the map, expressed as a fraction?

Qwen Base 7B — Miles-per-inch scale (Sample 7; correct).

```
To solve this problem, we need to determine the scale of the map and then
use that scale to find the distance between the two buildings on the map.
Here's a step-by-step approach:

1. **Determine the scale of the map:**
   - We know that 3 inches on the map represents 10 miles in real life.
```

```
   - Therefore, the scale of the map is \( \frac{10 \text{ miles}}{3
   \text{ inches}} \).

2. **Find the distance between the two buildings on the map:**
   - The real-life distance between the two buildings is 53.25 miles.
   - To find the map distance, we use the scale we found in step 1.
   - Let \( x \) be the distance on the map. Then, \( x \) inches on the
   map corresponds to 53.25 miles in real life.
   - Using the scale, we set up the equation: \( x \times \frac{10 \text{
   miles}}{3 \text{ inches}} = 53.25 \text{ miles} \).

3. **Solve for \( x \):**
   - Rearrange the equation to solve for \( x \): \( x = 53.25 \times
   \frac{3 \text{ inches}}{10 \text{ miles}} \).
   - Simplify the expression to get the value of \( x \).

Let's perform these calculations in Python to ensure accuracy.
```python
# Given values
map_scale_miles_per_inch = 10 / 3  # scale of the map in miles per inch
real_distance_miles = 53.25  # real distance between the buildings in
miles

# Calculate the distance on the map
map_distance_inches = real_distance_miles * (3 / 10)
print(map_distance_inches)
```
```output
15.975
```
The distance between the two buildings on the map is \(15.975 \) inches.
To express this as a fraction, we recognize that \(15.975 \) is the same
as \(\frac{15975}{1000} \). We can simplify this fraction by finding the
greatest common divisor (GCD) of 15975 and 1000.

Let's simplify the fraction.
```python
import math

# Values for the fraction
numerator = 15975
denominator = 1000

# Calculate the greatest common divisor (GCD)
gcd = math.gcd(numerator, denominator)

# Simplify the fraction
simplified_numerator = numerator // gcd
simplified_denominator = denominator // gcd

simplified_fraction = (simplified_numerator, simplified_denominator)
print(simplified_fraction)
```
```output
(639, 40)
```
The simplified fraction for \(15.975 \) is \(\frac{639}{40} \).

Therefore, the distance between the buildings on the map is
\(\boxed{\frac{639}{40}}\) inches.
```

Qwen SimpleRL 7B — Decimal decomposition (Sample 2; correct).

```
To solve this problem, we need to determine the distance on the map that
corresponds to a real-life distance of 53.25 miles, given that 3 inches
on the map equates to 10 miles in real life.

Step 1: Find the map distance for 1 mile in real life.
```

```
Since 3 inches on the map represent 10 miles, we can find out how many
inches on the map correspond to 1 mile by dividing 3 inches by 10 miles:
\[
\text{Map distance per real-life mile} = \frac{3 \text{ inches}}{10
\text{ miles}} = \frac{3}{10} \text{ inches per mile}
\]

Step 2: Calculate the map distance for 53.25 miles.
Now, we need to find out how many inches on the map correspond to 53.25
miles. We do this by multiplying the map distance per real-life mile by
53.25 miles:
\[
\text{Map distance for 53.25 miles} = 53.25 \text{ miles} \times
\frac{3}{10} \text{ inches per mile}
\]

Step 3: Perform the multiplication.
\[
53.25 \times \frac{3}{10} = \frac{53.25 \times 3}{10} = \frac{159.75}{10}
= 15.975 \text{ inches}
\]

Step 4: Express the answer as a fraction.
To express 15.975 as a fraction, we recognize that 15.975 is the same as
\(15 + 0.975\). We can convert 0.975 to a fraction:
\[
0.975 = \frac{975}{1000}
\]
We simplify \(\frac{975}{1000}\) by finding the greatest common divisor
(GCD) of 975 and 1000, which is 25:
\[
\frac{975 \div 25}{1000 \div 25} = \frac{39}{40}
\]
So, \(0.975 = \frac{39}{40}\). Therefore:
\[
15.975 = 15 + \frac{39}{40} = \frac{15 \times 40}{40} + \frac{39}{40} =
\frac{600}{40} + \frac{39}{40} = \frac{639}{40}
\]

Thus, the distance between the buildings on the map is
\(\boxed{\frac{639}{40}}\) inches.
```

OLMo-3 7B RLVR — Fraction conversion (Sample 0; correct).

```
Let's break down the problem step by step:

**Given:**
- 3 inches on the map = 10 miles in real life.

We are asked: If the real-life distance is 53.25 miles, what is the
corresponding distance on the map, in inches? And we need to express the
answer as a fraction.

---

### Step 1: Find the scale factor (miles per inch)

\[
10 \text{ miles} = 3 \text{ inches} \implies 1 \text{ inch} =
\frac{10}{3} \text{ miles}
\]

Alternatively, to find how many inches correspond to 1 mile:
\[
1 \text{ mile} = \frac{3}{10} \text{ inches}
\]

But it's often easier to set up a proportion.
```

```
---

### Step 2: Set up a proportion

Let \( x \) be the number of inches on the map for 53.25 miles.

\[
\frac{3 \text{ inches}}{10 \text{ miles}} = \frac{x \text{ inches}}{53.25
\text{ miles}}
\]

Cross-multiply:

\[
3 \times 53.25 = 10 \times x
\]
\[
159.75 = 10x
\]
\[
x = \frac{159.75}{10}
\]
\[
x = 15.975
\]

Now, we need to express this as a **fraction**, not a decimal.

Let's convert 15.975 to a fraction.

First, write the decimal as a fraction:

15.975

The decimal part is .975. Let's express that as a fraction.

.975 = 975/1000

Simplify 975/1000:
Divide numerator and denominator by 25:

975 ÷ 25 = 39
1000 ÷ 25 = 40

So .975 = 39/40

Thus,
15.975 = 15 and 39/40 = \(\frac{15 \times 40 + 39}{40} = \frac{600 +
39}{40} = \frac{639}{40}\)

So,

\[
x = \frac{639}{40}
\]

Let's double-check this calculation using the initial proportion method
to be sure.

Alternatively, using the scale factor:

Since 10 miles = 3 inches, then 1 mile = 3/10 inches.

So,
53.25 miles = 53.25 × (3/10) inches

Convert 53.25 to a fraction:
```

```
53.25 = 53 + 1/4 = (53 × 4 + 1)/4 = (212 + 1)/4 = 213/4

So,

53.25 × (3/10) = (213/4) × (3/10) = (213 × 3) / (4 × 10) = 639 / 40

Same result!

---

### Final Answer

\[
\boxed{\dfrac{639}{40}}
\]
```

J Training and Evaluation Details

Hardware and software. All PPO training and all evaluations ran on a Linux server with Ubuntu 24.04 LTS, two Intel Xeon Gold 6330 CPUs, 503 GiB RAM, and two NVIDIA GeForce RTX 4090 GPUs. The stack is Python 3.10.20, PyTorch 2.4.0 with CUDA 12.1, Transformers 4.57.1, vLLM 0.6.3, Ray 2.55.1, and VERL 0.1.

PPO training configuration. The Qwen2.5-3B Countdown run follows the TinyZero setup (Pan et al., 2025) with PPO (Schulman et al., 2017). The actor uses AdamW (Loshchilov & Hutter, 2019) with constant learning rate 10^{-6}, 160 prompts per iteration, mini-batch size 64, and micro-batch size 4; the critic learning rate is 10^{-5}. Rollouts use temperature 1.0, top-$p = 1.0$, and a 1,024-token response limit, while Countdown evaluation uses a 256-token limit. GAE (Schulman et al., 2018) uses $\gamma = 1.0$ and $\lambda = 1.0$, rollout group size is $G = 1$, and the actor KL loss is disabled. Training runs 15 epochs and 276 optimizer steps indexed 0 to 275, with seed 1 and checkpoints every 25 steps.

K Extended Related Work

RLVR, empirical support, and reasoning boundaries. Modern RLVR systems build on outcome-based optimization in the GRPO and PPO families (Shao et al., 2024; Guo et al., 2025; Yu et al., 2025; Zeng et al., 2025). Large-k evaluations identify regimes where RL improves sampling efficiency without preserving the base model's empirical answer support (Yue et al., 2025; Wu et al., 2026), while process-aware evaluation and prolonged exploratory RL report expansion in others (Wen et al., 2026; Liu et al., 2025b). Further analyses tie contraction to amplification of pretraining-favored modes, winner-take-all dynamics, and finite on-policy exposure (Zhao et al., 2025; Nguyen et al., 2025; Yuan et al., 2026; Zhou, 2026). These works ask which prompts or answers remain reachable. We ask where probability goes inside a prompt that stays solvable.

Diversity-preserving optimization. Training-time methods counter sharpening by reweighting unlikely correct rollouts, smoothing updates, changing divergence geometry, or reallocating exploration (He et al., 2025; Gai et al., 2025; Li et al., 2026a; Song et al., 2025; Yao et al., 2025; Hu et al., 2026). Our entrance-level analysis supplies a concrete target for them: probability mass over alternative early solution families.

Forking decisions and prefix interventions. Token-level work identifies sparse decisions with outsized downstream influence (Bigelow et al., 2025; Wang et al., 2025; Kim & No, 2026), and BODHI measures semantic branching from sampled traces (Saha et al., 2026). Prefix methods move the states from which rollouts begin (Zhang et al., 2025; Macar et al., 2026). Entrance families differ in two ways: they partition the complete solver-enumerated valid solution set, including families absent from model samples, and their entrances can be supplied without copying a successful trace.

Inference-time breadth and checkpoint composition. Repeated sampling, self-consistency, Best-of-N, and verifier-guided search exploit only modes that remain reachable under the sampling

policy (Brown et al., 2024; Wang et al., 2023; Huang et al., 2025; Snell et al., 2025). Sampling across training checkpoints and weight-space interpolation recover modes that fade during post-training (Li et al., 2026c;b; Dang et al., 2025). Our results tie those methods to a location: they work when they restore diversity at the states where solution families are chosen.

Measuring reasoning breadth. Recent metrics separate process correctness, breadth, depth, stability, and semantic path diversity (Wen et al., 2026; Dragoi et al., 2025; Liu et al., 2025a; Ju et al., 2026; Yu, 2025). On Countdown, exhaustive enumeration supplies a complete denominator, the canonical solution set with its opening families, which is what makes failure to enter a family distinguishable from failure to complete it once entered.