DIVERSITY COLLAPSE IN RLVR: EVEN GOOD STRATEGIES DIE IF THEY START RARE

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) improves reasoning accuracy, but it can also narrow the set of strategies a model uses. We ask which strategies survive this process and why. Across a full RLVR trajectory on Countdown and three base/RL model pairs on MATH-500, we find a consistent rich-get-richer effect: strategies that are already common become more dominant after training, while rare strategies are much more likely to disappear. Strategy quality provides weaker and less consistent protection; even reliable strategies can be lost when they begin uncommon. On Countdown, the effective number of strategies per problem falls from about 3 to nearly 1, while problems whose strategies start at similar frequencies retain substantially more diversity. We further find that finite training samples can amplify this imbalance; once a strategy becomes sufficiently rare, it may stop appearing in sampled rollouts and receive no direct reinforcement. These results show that RLVR can favor what is already common over what is merely effective, causing early frequency differences to compound into the loss of useful reasoning strategies.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) improves mathematical and logical reasoning in large language models (Shao et al., 2024; Guo et al., 2025; Yu et al., 2025; Zeng et al., 2025), but it can also reduce the variety of solutions they produce (Yue et al., 2025; Wu et al., 2025; Cui et al., 2025). This loss goes beyond wording differences. After training, models may rely on fewer reasoning strategies (Saha et al., 2026). On Countdown, Zhou & Li (2026) find that RLVR-trained models often stop choosing certain first steps even though they can still finish those solutions when the missing first step is supplied.

These findings establish that reasoning diversity can shrink, but they leave open a more specific question: **what determines which particular strategies survive?** Existing work suggests two possibilities. RLVR may preserve strategies that work well, since successful behaviors receive stronger reinforcement and rare high-quality behaviors may recover during training (Yao et al., 2025; Fan et al., 2026). Alternatively, survival may depend strongly on how common a strategy already is. Strategies that appear more often are sampled more often and receive more opportunities for reinforcement, allowing an early frequency advantage to compound (Zhao et al., 2025; Fan et al., 2026). Under this view, even a highly successful strategy can disappear simply because it starts rare.

Prior work provides evidence for both mechanisms, but does not directly compare them at the level of identifiable strategies competing on the same problem. In particular, it does not ask whether the survival of a specific reasoning strategy is better predicted by how often the model initially uses it or by how reliably that strategy succeeds. This is the question we study. We track the same reasoning strategies across training and measure their frequency, quality, and whether they remain observable at later checkpoints.

To do this, we group responses into **strategy families**, sets of solutions that begin with the same basic approach. A family's **frequency** q is the fraction of sampled responses that use it, while its **quality** μ is the fraction of those responses that are correct. We follow these families across six checkpoints from one Countdown training run (where the model combines given numbers to reach a target value), then test whether the same overall pattern appears in three MATH-500 base/RL model pairs (models before and after RLVR on mathematical reasoning problems).

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.

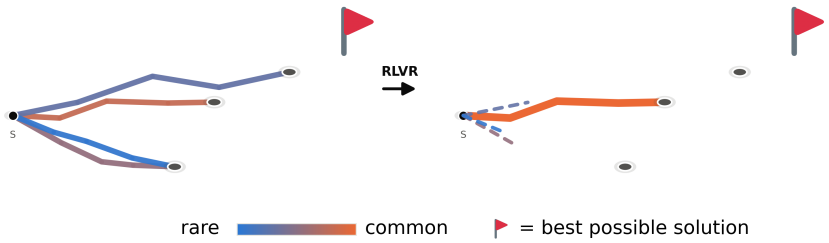

Figure 1: Illustration of strategy selection under RLVR. Several strategies are used at the start. Training can increase the share of an already common strategy while less common alternatives become difficult to observe, even when they can also solve the problem.

We find that frequency is the more consistent predictor of strategy survival. Strategies that are already common become increasingly dominant relative to the starting model, while rare strategies are much more likely to disappear from our samples even when they have high success rates. Quality provides additional signal, but its effect is weaker and less consistent. Problems whose strategies begin at similar frequencies also retain substantially more diversity, showing that the initial balance between strategies matters for what survives later.

These results sharpen the existing picture of RLVR diversity collapse. The contribution is not simply that common behaviors can be amplified, but that, among identifiable reasoning strategies for the same problem, starting frequency predicts survival more consistently than measured strategy quality. We also examine one mechanism that may amplify this advantage. As a strategy becomes rare, it becomes more likely to be absent from a finite rollout group and therefore receive no direct reinforcement in that update. A simplified simulation incorporating this effect accounts for part of the concentration we observe on Countdown. The same frequency advantage appears across all three MATH-500 model pairs, showing that the pattern extends beyond Countdown, while the full Countdown trajectory reveals how this advantage can compound into much stronger strategy loss over training.

2 Related Work

RLVR and the reasoning boundary. Recent work asks whether RLVR creates new reasoning abilities or mainly changes which existing behaviors the model chooses. The evidence is mixed. Yue et al. (2025) show that RLVR improves pass@1, or single-attempt accuracy, while the base model can solve more problems when both models receive many attempts. Other work also finds that RLVR can narrow the range of answers a model produces (Wu et al., 2025). In contrast, ProRL reports that longer RL training can produce correct solutions not observed from the base model (Liu et al., 2025). Yao et al. (2025) propose that both patterns can occur at different stages, with early training favoring already common successful behaviors and later training allowing rare, higher-quality behaviors to recover. We ask a narrower question. Among reasoning strategies already visible before or early in RLVR, which ones remain visible as training continues?

Strategy collapse and selection. RLVR can reduce the number of distinct reasoning approaches a model uses, not only change how answers are written (Saha et al., 2026). Zhou & Li (2026) show that many Countdown strategies stop appearing at their first arithmetic step, even though the model can still complete them when that step is supplied. Zhao et al. (2025) likewise find that RL can increase reliance on behaviors that were already common. One explanation is sampling. Rare strategies appear less often in RL rollouts and therefore receive fewer direct opportunities for reinforcement (Fan et al., 2026). Continued training can also keep narrowing behavior after a problem is already solved reliably (Yuan et al., 2026; Zhou, 2026). In contrast, Yao et al. (2025) argue that frequency advantages can weaken later, allowing rare high-quality behaviors to recover. We test these competing patterns by tracking identifiable strategy families through training.

Measuring and preserving diversity. Other work develops ways to measure or preserve reasoning diversity. Reasoning Path Divergence measures differences between reasoning steps across solutions (Ju et al., 2025). Proposed training methods include giving more weight to rare correct solutions (He et al., 2025), changing the pass@k objective (Chen et al., 2025), rewarding novel behaviors (Tuyls et al., 2026), and encouraging different solution approaches during sampling (Kang & Zhang, 2025). Our goal is different. We do not introduce a diversity-preserving method. We use standard RLVR runs to study how existing frequency differences relate to later concentration and disappearance.

3 METHOD

3.1 STRATEGIES, FREQUENCY, AND QUALITY

For each problem, we group responses into strategy families, meaning solutions that begin with the same basic approach. Let x be a problem and π_t the model at checkpoint t. For a strategy family b, we measure

$$q_t(b \mid x) = \Pr_{y \sim \pi_t(\cdot|x)}[y \in b], \qquad \mu_t(b \mid x) = \Pr[y \text{ correct} \mid y \in b]. \tag{1}$$

Here, q_t is how often the model uses the strategy at checkpoint t, and μ_t is its success rate. When the checkpoint is clear, we write q and μ. We use q_0 for the starting distribution, q_t for a later checkpoint, and $\hat{q}_t$ for a model prediction. For endpoint comparisons, q_0 and q_t correspond to the base and RL models.

Quality estimates from very small families can be unstable. We therefore use

$$\mu = \frac{n_{\text{correct}} + 1}{n_{\text{total}} + 2}, \tag{2}$$

which adds one hypothetical success and one hypothetical failure. This keeps an estimate based on only a few observations from becoming exactly 0 or 1.

We use two comparison types. A **trajectory** follows one training run across checkpoints and can show when concentration appears. An **endpoint pair** compares a base model with an RL-trained model and shows only the total change.

Countdown has a finite set of arithmetic solutions. Following Zhou & Li (2026), we define a family by its *entrance*, the first arithmetic operation in the response. For example, solutions beginning with "$4\times$" share a family even if later steps differ. We extract this operation from the `<think>` block using their released parser.

For MATH-500, strategies cannot be listed in advance. We use the first complete calculation in a response as a marker of its opening approach. We normalize formatting, numeric notation, and operand order for commutative operations, then group responses whose normalized first calculations match exactly. The parser covers 75.5–76.9% of responses. Full rules and limitations appear in Appendix A. We retain families observed at least 5 times in the base model and problems with at least 2 such families.

3.2 MODELS OF STRATEGY SELECTION

We compare three simple explanations for how a starting strategy distribution becomes a later distribution. Let $q_0(b)$ be a strategy's starting frequency and $q_t(b)$ its predicted frequency at a later checkpoint or RL endpoint.

$$\text{Frequency (Model A)} \quad q_t(b) \propto q_0(b)^{\alpha}, \tag{3}$$

$$\text{Quality (Model B)} \quad q_t(b) \propto q_0(b) \exp(\mu_0(b)/\beta)\,, \tag{4}$$

$$\text{Frequency + quality (Model C)} \quad q_t(b) \propto q_0(b)^{\alpha} \exp(\mu_0(b)/\beta)\,. \tag{5}$$

Model A asks whether existing frequency differences are amplified. When $\alpha > 1$, common strategies gain share relative to rare strategies. Model B asks whether higher-quality strategies gain share. Model C allows both signals to help. In the appendix, α_A refers to the frequency parameter from Model A, while α_C and $1/\beta_C$ refer to Model C.

For the Countdown trajectory, every later checkpoint is fit relative to step 25. The reported α is therefore *cumulative*, not a per-update selection rate. A larger value at a later checkpoint means that the distribution has become more strongly related to the initial frequency imbalance. It does not by itself show that each individual training update applies stronger selection than earlier updates.

These smooth models can make a strategy extremely rare but never set its probability to zero. We separately study a simple finite-sampling model. The released Countdown training setup samples $G = 8$ responses for each prompt in an update. In our simulation, a family that receives zero of those G samples is removed from later simulated steps. Permanent removal is a modeling assumption used to test whether limited sampling could amplify concentration. It should not be read as a literal claim that one missed real training group permanently deletes a behavior from the network.

3.3 Analysis

For Countdown, we define families at step 25 and keep this set fixed. A family remains in the analysis even when it is not observed at a later checkpoint. Of the 1000 sampled Countdown problems, 969 have at least two families observed at least 5 times at step 25. These 969 problems form the main Countdown analysis set.

Change in frequency. We measure each family's change from a starting distribution to a later distribution as

$$\delta_t(b) = \log(q_t(b) + \epsilon) - \log(q_0(b) + \epsilon), \qquad \epsilon = 1/512. \tag{6}$$

We use $1/512$ only as a small numerical floor for the logarithm. It is below the resolution of the 64-sample Countdown estimates and does not represent an observed probability. Appendix B reports the regression details and sensitivity checks. Predictors are standardized within each problem so strategies are compared with other strategies competing on the same problem.

Observed disappearance. For a Countdown transition from checkpoint t to $t+1$, we call a family **unobserved** at $t+1$ when it appears at t but receives zero counts among the 64 evaluation samples at $t+1$. Throughout this paper, disappearance refers only to this empirical sampling event. It means below detection at $n = 64$, not proven zero probability under the policy. We fit a logistic regression predicting this event from the family's frequency and quality at checkpoint t. Both predictors are standardized within each problem.

Overall diversity. We summarize how broadly probability is spread across families using the effective number of strategies

$$N_{\text{eff}}(q) = \exp(H(q)), \tag{7}$$

where $H(q)$ is entropy. N_{eff} is near 1 when one strategy dominates and grows when probability is spread more evenly.

Predicting concentration. Using the fitted Model A parameter, we compute a predicted later distribution $\hat{q}_t(b) \propto q_0(b)^\alpha$ and its effective number of strategies. We compare predicted with observed diversity across problems using a calibration slope. A slope near 1 means the predicted differences across problems have roughly the observed scale. A much smaller slope means the model predicts less concentration than we observe. We repeat this analysis with the finite-sampling simulation.

Problems that resist concentration. We call a Countdown problem **resistant** when at least two of its step-25 qualifying families are still observed at every later checkpoint. Here "observed" means at least one of the 64 evaluation samples, not at least five. Of the 969 analysis problems, 224 meet this definition. We predict resistance using only step-25 information. Features include the number of observed qualifying families, how evenly their probabilities are spread, the largest family share, family quality, base accuracy, and the number of entrance families that the exact Countdown solver shows can lead to a valid solution.

Model fitting and uncertainty. For each problem, we observe counts for its strategy families and choose parameters that make those counts most likely. If n_{xb} is the observed count for family b on

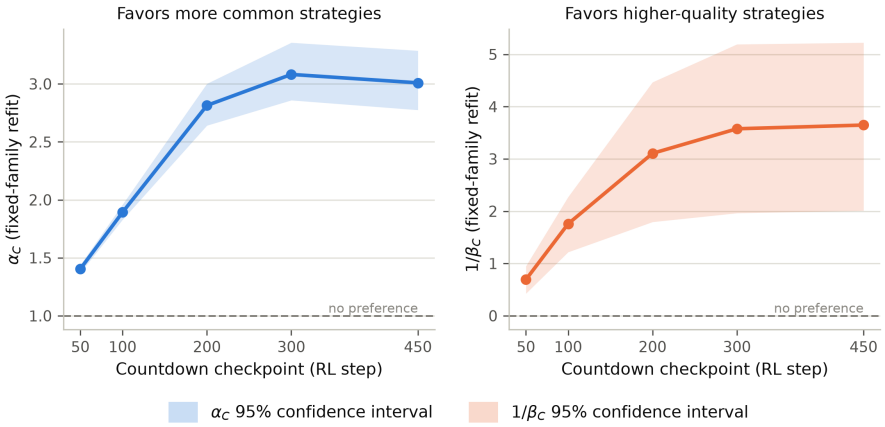

Figure 2: Cumulative frequency and quality parameters from Model C across Countdown checkpoints, each fit relative to step 25. Shaded bands show 95% confidence intervals from problem-level bootstrap resampling.

problem x, and $p_{xb}(\theta)$ is its probability under a selection model, we maximize

$$\ell(\theta) = \sum_x \sum_{b \in \mathcal{B}(x)} n_{xb} \log p_{xb}(\theta). \tag{8}$$

For Model A, $\theta = \alpha$. For Model B, $\theta = 1/\beta$. For Model C, $\theta = (\alpha, 1/\beta)$. We optimize with L-BFGS-B under

$$0.01 \le \alpha \le 20, \qquad -20 \le 1/\beta \le 20. \tag{9}$$

We compare models with AIC, BIC, likelihood-ratio tests for nested models, and five-fold cross-validation that holds out entire problems. Confidence intervals use 1000 bootstrap resamples of whole problems.

3.4 Experimental Setup

Our main trajectory uses the public GRPO checkpoint series `philschmid/qwen-2.5-3b-r1-countdown`, based on Qwen2.5-3B-Instruct. We analyze checkpoints 25, 50, 100, 200, 300, and 450, treating step 25 as the starting distribution.

For mathematical reasoning, we evaluate three SimpleRL-Zoo base/RL endpoint pairs, one math-specialized 7B model and general-purpose 1.5B and 7B models.

For MATH-500, we sample 256 responses for each of 100 problems at temperature 1.0, top-p 1.0, and maximum generation length 2048. For Countdown, we sample 64 responses for each of 1000 problems at each checkpoint, using temperature 1.0 and maximum generation length 1024.

Countdown answers are checked with an exact solver. For MATH-500, we normalize the final boxed answer and compare it with the reference. When direct comparison is inconclusive, we use SymPy to test symbolic equivalence. Exact model revisions, seeds, decoding settings, optimization details, and compute appear in Appendix D.

4 Experiments

We first use the Countdown trajectory to follow strategy selection through training. We then ask whether the same frequency advantage appears across three MATH-500 base/RL pairs.

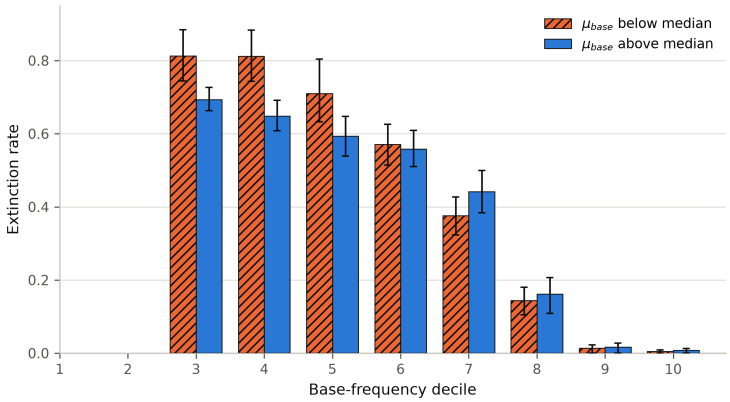

Figure 3: Observed disappearance during the step-100 $\rightarrow$ 200 transition falls sharply with strategy frequency. A family disappears when it is present at step 100 but absent from all 64 samples at step 200. Within similar frequency ranges, higher-quality strategies show little consistent survival advantage. Error bars are 95% bootstrap confidence intervals.

Table 1: Predictors of strategy extinction across Countdown training. Negative coefficients mean that higher values reduce the probability of extinction. Bold indicates that the 95% confidence interval excludes zero.

Transition	Extinct / Total	Extinction rate	Frequency effect (95% CI)	Quality effect (95% CI)
25 $\rightarrow$ 50	85/3218	2.6%	$\mathbf{-1.46}$ $(-1.69, -1.28)$	$\mathbf{-0.28}$ $(-0.46, -0.10)$
50 $\rightarrow$ 100	332/3133	10.6%	$\mathbf{-1.55}$ $(-1.69, -1.44)$	-0.05 $(-0.15, 0.05)$
100 $\rightarrow$ 200	1168/2837	41.2%	$\mathbf{-2.16}$ $(-2.27, -2.07)$	-0.07 $(-0.16, 0.02)$
200 $\rightarrow$ 300	432/1684	25.7%	$\mathbf{-2.80}$ $(-3.08, -2.57)$	$\mathbf{0.22}$ $(0.09, 0.35)$
300 $\rightarrow$ 450	69/1264	5.5%	$\mathbf{-2.58}$ $(-3.25, -2.11)$	-0.09 $(-0.34, 0.16)$

4.1 Initial Frequency Advantage Compounds Over Training

Figure 2 asks how strongly later strategy frequencies are related to frequency and quality at the starting checkpoint. Because every fit compares a later checkpoint with step 25, the parameters summarize accumulated change from the same starting distribution.

The cumulative frequency parameter rises quickly. α_C increases from 1.40 at step 50 to 1.89 at step 100 and 2.81 at step 200, then remains near 3 at steps 300 and 450. Since $\alpha > 1$ amplifies existing frequency differences, later distributions become progressively more concentrated around strategies that were common near the start. This is evidence of compounding path dependence, not evidence that the per-update selection coefficient itself increases over time.

Model C also estimates a positive quality parameter at every checkpoint. The combined model fits better than frequency-only or quality-only alternatives in the MATH-500 comparisons reported later. However, the strategy-level change regressions in Appendix E show that the independent quality coefficient is small or unstable across much of the trajectory. We therefore treat quality as additional predictive signal rather than as a second effect as consistent as frequency.

The main pattern is simpler. Starting common gives a strategy an advantage that becomes increasingly visible as training proceeds. We next ask whether the same frequency advantage predicts the sharper event of no longer being observed.

4.2 Rare Strategies Are More Likely to Disappear from Samples

We next ask what predicts whether a strategy becomes unobserved. Figure 3 focuses on the step-100 $\rightarrow$ 200 transition, when the largest number of currently observed families disappear from the 64-sample evaluation. Families are grouped by their frequency immediately before the transition.

The relationship is strong. Around 80% of qualifying families in the lower displayed frequency groups disappear. The rate falls steadily as frequency increases, dropping below 20% in high-frequency groups and approaching zero for the most common families. Because these events are measured from 64 samples, disappearance means that a family falls below our detection in that sample, not that its true model probability is exactly zero.

The same directional pattern appears across all five transitions. Table 1 reports logistic regressions using current frequency and quality together. The frequency coefficient is negative and its confidence interval excludes zero at every transition. Its magnitude ranges from -1.46 for step $25 \rightarrow 50$ to -2.80 for step $200 \rightarrow 300$. Thus, among strategies competing on the same problem, the rarer ones are consistently more likely to become unobserved at the next checkpoint.

Quality does not show the same consistency. Its coefficient differs from zero in only two of five transitions, and those two estimates point in opposite directions. Splitting families by quality within similar frequency ranges also produces much smaller differences than moving across frequency groups. A strategy can therefore have a high measured success rate and still disappear from sampled behavior when its current frequency is low.

This result connects the paper's title to the training dynamics. Strategies that start rare are more exposed to early losses in probability, and once they become rare during training, current frequency strongly predicts whether they remain observable. Appendix E gives concrete cases where a less common but more reliable family disappears while a more common, less reliable family remains.

4.3 Initially Balanced Problems Resist Concentration

The frequency story makes another prediction. If several strategies begin at similar frequencies, none has a large initial advantage to amplify. We test this on the 969 Countdown problems that have at least two qualifying families at step 25. Of these, 224 remain resistant, meaning that at least two of those families are observed at every checkpoint.

A model using only step-25 features predicts resistance with AUC 0.702, with a 95% confidence interval from 0.679 to 0.733. The clearest predictor is initial entropy, which measures how evenly probability is spread across strategies. Problems with more even starting distributions are more likely to retain several observed strategies. Simply having more qualifying families is not enough after this evenness is taken into account.

Quality-related features do not reliably separate resistant from non-resistant problems. Mean quality, variation in quality, and base-model accuracy all have confidence intervals that include zero. We also include the number of solver-valid entrance families, meaning first-step families that the exact Countdown solver shows can lead to at least one correct solution. Its coefficient also includes zero. The strongest signal is therefore the initial shape of the frequency distribution.

Resistant problems do not show a clear final accuracy penalty. They begin about 2.2 percentage points less accurate at step 25, but by step 450 the accuracy difference is not distinguishable from zero. Both groups improve by roughly 35 points. Preserving several observed strategies is therefore not obviously associated with worse final accuracy in this run.

4.4 Finite Sampling Explains Part of the Extra Collapse

The results so far show that rare families become increasingly difficult to observe. We next ask whether smooth frequency reweighting alone can reproduce how much diversity is lost. Figure 4 compares predicted and observed effective strategy counts. A calibration slope of 1 means that predicted variation across problems has approximately the observed scale.

The smooth q^{α} model works much better early than late. Its slope is about 0.78 at step 50, then falls to roughly 0.51 at step 100, 0.18 at step 200, and 0.06 by step 450. The model therefore captures part of early concentration but increasingly underpredicts the sharpness of later collapse.

We then add the finite-sampling simulation described in Section 3.2. At each simulated checkpoint, we reweight the surviving families from the step-25 reference, draw G responses, remove families receiving zero draws under this simplified assumption, renormalize, and continue. The number of simulated update stages matches the sequence of observed Countdown checkpoints. This construc-

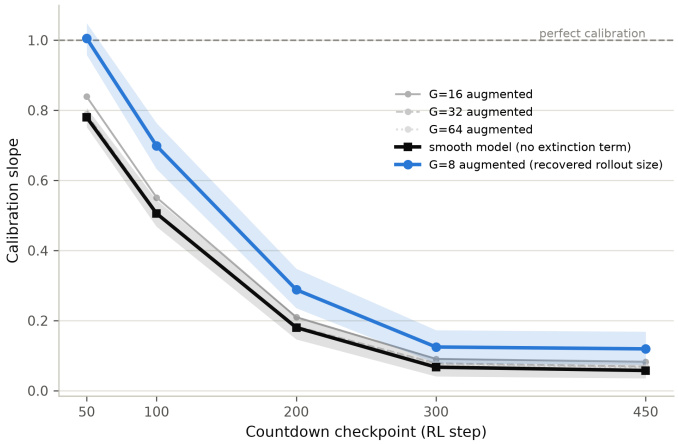

Figure 4: Adding a simplified finite-sampling removal step improves predicted concentration, especially at the actual rollout group size $G = 8$, but still does not explain the full late-stage collapse.

Table 2: Strategy selection across Countdown and MATH-500. $\alpha > 1$ means that initially common strategies become more favored; larger $1/\beta$ means a stronger quality effect. Effective families report strategy diversity before and after RLVR.

Setting	α (95% CI)	$1/\beta$ (95% CI)	Extinction	Effective families
Countdown $25 \to 50$	$1.40\ (1.37, 1.44)$	$0.69\ (0.43, 0.95)$	2.6%	$2.91 \to 2.43$
Math-7B	$1.10\ (1.05, 1.17)$	$0.32\ (0.08, 0.55)$	2.6%	$4.24 \to 3.72$
Qwen2.5-1.5B	$1.27\ (1.18, 1.35)$	$0.96\ (0.31, 1.59)$	3.7%	$3.74 \to 3.04$
Qwen2.5-7B	$1.10\ (1.05, 1.16)$	$0.76\ (0.40, 1.08)$	1.2%	$4.10 \to 3.59$

tion intentionally exaggerates the permanence of one missed training group, but it provides a test of whether finite rollout groups can move concentration in the observed direction.

At $G = 8$, the calibration slope reaches about 1 at step 50 instead of 0.78 under the smooth model. It also improves at every later checkpoint, rising from about 0.18 to 0.29 at step 200 and from about 0.06 to 0.12 at step 450. Larger values of G move the simulation back toward the smooth model because rare families are less likely to receive zero samples.

Finite sampling therefore explains part, not all, of the extra concentration. The $G = 8$ simulation remains far below perfect calibration late in training. The empirical data show a related pattern. More families fall below the rough $1/G$ frequency scale as training proceeds, and families that become unobserved later in training are less likely to reappear in subsequent 64-sample evaluations. We describe this as persistent non-observation rather than proof that their policy probability becomes exactly zero.

The practical distinction is important. If training only lowers a strategy's probability, broader decoding may reveal it again. If repeated updates drive it below the range sampled during ordinary use or training, recovery becomes harder. Our results support limited sampling as one contributor to this process, while leaving room for other mechanisms created by shared model parameters and optimization dynamics.

4.5 The Frequency Effect Replicates on Mathematical Reasoning

Finally, we ask which parts of the story extend beyond Countdown. Table 2 compares the first Countdown transition with three MATH-500 base/RL pairs, a math-specialized 7B model and general-purpose 1.5B and 7B models.

The frequency effect appears in all three MATH-500 settings. The fitted Model C frequency parameter α_C is 1.10 for the math-specialized 7B model, 1.27 for the general 1.5B model, and 1.10 for the

general 7B model, with all three confidence intervals above 1. Strategies that begin more common therefore gain relative share after RLVR across all three model pairs.

Model C also estimates a positive quality parameter in each MATH-500 pair, and the combined model has better likelihood-based fit than either single-factor model. This does not conflict with the near-zero quality coefficients in some continuous-change regressions. The two analyses ask different questions. Model C asks whether quality helps explain the full normalized distribution jointly with frequency, while the change regression asks for a linear within-problem association with noisy strategy-level log changes. We therefore conclude only that quality adds some distribution-level predictive information, not that it has a large or stable independent effect on every family's change.

The size of diversity loss is much smaller than in late Countdown. Countdown falls from 2.91 effective strategies at step 25 to 1.15 at step 450, a reduction of about 60%. The three MATH-500 endpoint pairs lose about 12–19%. Observed disappearance is also rare on MATH-500, between 1.2% and 3.7% under the endpoint sampling definition.

The replication therefore supports the broadest part of the story, RLVR favors strategies that were already common. The stronger claim about widespread, persistent non-observation is supported only by the full Countdown trajectory. MATH-500 provides too few disappearing families for a reliable two-predictor disappearance regression.

Across the three MATH-500 pairs, the parser assigns a strategy family to 75.5–76.9% of responses, yielding 378–485 qualifying families across 78–88 problems. Full parser validation appears in Appendix A.

5 Conclusion

We study what determines which reasoning strategies survive RLVR training. By tracking identifiable strategy families across six checkpoints of a Countdown run, we compare how often a strategy is used with how reliably it succeeds, then test the same frequency pattern across three MATH-500 base/RL model pairs.

Three findings emerge. First, starting frequency is the most consistent predictor of strategy survival. Strategies that begin common become increasingly dominant relative to the starting model, while rare strategies are much more likely to disappear from our samples even when they have high measured success rates. Second, problems whose strategies begin at similar frequencies retain substantially more diversity, showing that the initial balance between strategies matters for what survives later. Third, limited sampling can amplify this process: a simplified finite-sampling simulation explains part of the concentration that smooth frequency reweighting alone does not capture. The same frequency advantage appears across all three MATH-500 model pairs, while the full Countdown trajectory shows how this advantage can compound into stronger strategy loss over training.

Together, these results sharpen the picture of diversity collapse under RLVR. Among strategies already available to the model, how often a strategy is used can be more predictive of whether it survives than how reliably it succeeds. Small initial frequency differences can therefore grow until useful but uncommon strategies are rarely observed.

Broader impact. Understanding which reasoning strategies survive RLVR may help guide training methods that improve accuracy without unnecessarily narrowing the model's available approaches. Preserving useful but uncommon strategies could improve robustness and maintain alternative ways of solving a problem, although greater diversity is not always beneficial because uncommon strategies may also be incorrect or undesirable.

Limitations and future work. Our strongest evidence that rare strategies stop being observed over training comes from Countdown, where absence from 64 sampled responses does not imply zero true probability. The finite-sampling analysis captures only one possible mechanism. Future work should causally vary the starting frequency of rare, reliable strategies, test longer trajectories and other RLVR settings, and study whether explicitly preserving rare strategies improves robustness or performance.

AI USE STATEMENT

Generative AI tools were used to revise and polish manuscript text, assist with literature discovery and organization, and support discussion of experimental design, analysis, and presentation. They were also used for LaTeX and formatting assistance. All AI-assisted research decisions, citations, analyses, and manuscript content were reviewed and verified by the authors.

ETHICS STATEMENT

This work analyzes publicly available language models and reasoning benchmarks and does not involve human subjects or private data. Our results characterize changes in observed reasoning strategies under RLVR, but absence from a finite sample does not establish that a strategy is completely unavailable to the model. We therefore distinguish observed disappearance from true loss of a strategy and avoid treating our analysis as evidence of permanent model-level deletion. More broadly, understanding how RLVR changes reasoning diversity may help identify training procedures that preserve useful but uncommon behaviors.

REPRODUCIBILITY STATEMENT

The main text and appendices describe the models, datasets, sampling settings, strategy-family construction, evaluation procedures, statistical analyses, simulations, and uncertainty estimates used in our experiments. We report model revisions, random seeds, decoding settings, inclusion thresholds, and optimization details. Code, experiment configurations, cached analysis outputs, and scripts for reproducing the reported results and figures will be released at `https://anonymous.4open.science/r/diversity-collapse-rlvr/`. All linked materials are anonymized for double-blind review.

REFERENCES

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Mingyuan Fan, Weiguang Han, Daixin Wang, Cen Chen, Zhiqiang Zhang, and Jun Zhou. When sharpening becomes collapse: Sampling bias and semantic coupling in rl with verifiable rewards. *arXiv preprint arXiv:2601.15609*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Andre He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*, 2025.

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. *arXiv preprint arXiv:2510.26122*, 2025.

Shijia Kang and Muhan Zhang. The road less traveled: Enhancing exploration in llms via sequential sampling. *arXiv preprint arXiv:2510.15502*, 2025.

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. *arXiv preprint arXiv:2505.24864*, 2025.

Soumadeep Saha, Krish Sharma, Akshay Chaturvedi, and Nicholas Asher. BODHI: Do llms branch out and discover heterogeneous inferences? *arXiv preprint arXiv:2608.02867*, 2026.

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. *arXiv preprint arXiv:2402.03300*, 2024.

Jens Tuyls, Dylan J. Foster, Akshay Krishnamurthy, and Jordan T. Ash. Representation-based exploration for language models: From test-time to post-training. In *International Conference on Learning Representations*, 2026.

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. *arXiv preprint arXiv:2507.14843*, 2025.

Xinhao Yao, Lu Yu, Xiaolin Hu, Fengwei Teng, Qing Cui, Jun Zhou, and Yong Liu. The debate on rlvr reasoning capability boundary: Shrinkage, expansion, or both? a two-stage dynamic view. *arXiv preprint arXiv:2510.04028*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, et al. DAPO: An open-source llm reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, 2025.

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. *arXiv preprint arXiv:2606.15455*, 2026.

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 *Advances in Neural Information Processing Systems*, volume 38, 2025.

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 *Conference on Language Modeling*, 2025.

Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. In *Conference on Language Modeling*, 2025.

Qiancheng Zhou and Ruizhe Li. Locked at the entrance, open inside: Where rlvr narrows the solution space. *arXiv preprint arXiv:2608.29188*, 2026.

Todd Zhou. When rlvr shrinks the reasoning boundary: Diagnosing pass@k inversion. *arXiv preprint arXiv:2607.20543*, 2026.

A STRATEGY-FAMILY CONSTRUCTION

A.1 COUNTDOWN ENTRANCES

A Countdown problem gives three or four numbers and a target integer. A valid solution uses each supplied number exactly once with arithmetic operations to reach the target.

Following Zhou & Li (2026), we group solutions by their *entrance*, the first operand and operator used in the model's reasoning. A response that begins by multiplying 4 belongs to the family '$4\times$,' while one that begins by subtracting from 61 belongs to '$61-$.' The entrance describes only the start of the solution. Responses in the same family can take different later steps.

We use three components released by Zhou & Li (2026). The exact solver `evaluate_native_countdown` checks whether a response reaches the target using the supplied numbers and allowed operations. The parser `find_first_reasoning_entrance` extracts the first operand-operator pair from the `<think>` block. The function `enumerate_solution_set` lists all valid canonical solutions. We use this enumeration to count **solver-valid entrance families**, meaning entrance families for which at least one exact Countdown solution exists.

For example, consider the problem $65, 64, 94 \rightarrow 35$. A response beginning

> First, let's consider how we can use the numbers 65, 64, and 94 to get to 35. The operation that might help us do this is subtraction or division as we are moving towards 35 from either a larger number...

is assigned to the family $94-$ because its first extracted operand is 94 and its first extracted operation is subtraction.

A.2 MATHEMATICAL-REASONING FIRST CALCULATIONS

For MATH-500, there is no finite list of all possible strategies. We therefore use the first complete calculation in a response as an observable marker of its opening approach.

The parser searches for the earliest arithmetic or algebraic equality with a concrete numerical result. It considers candidate "=" signs while ignoring comparison operators. The left-hand side must contain either at least two operands joined by an arithmetic operator or a coefficient-variable term such as $5G$, and the right-hand side must contain a concrete number.

We then put equivalent-looking calculations into a common text format before comparing them. This canonicalization removes LaTeX formatting, converts fractions to a standard form, normalizes numbers and whitespace, and sorts terms for commutative operations such as addition. Two responses are assigned to the same family when their normalized first calculations match exactly.

Example, removing formatting. A response containing

```
\(\frac{4.5 \text{ miles}}{1.25 \text{ hours}} = 3.6\)
```

is extracted and normalized to

```
(4.5)/(1.25) = 3.6.
```

The normalization removes mathematical delimiters and units and writes the fraction in a common form.

Example, sorting operands. The calculation

```
2 + 2 + 2 + 0 = 6
```

is normalized to

```
0 + 2 + 2 + 2 = 6.
```

For the same reason, `3 + 1 = 4` and `1 + 3 = 4` are placed in the same family.

Table 3: Parser coverage and resulting analysis units on MATH-500.

Setting	Overall	Base	RL	Families	Problems
Math-7B (specialized)	76.0%	–	–	470	83
Qwen2.5-1.5B (general)	76.9%	73.0%	80.7%	378	78
Qwen2.5-7B (general)	75.5%	73.5%	77.5%	485	88

A.3 Parser validation and limitations

Neither parser was evaluated against a full human-labeled benchmark. The Countdown parser is a deterministic regular-expression parser released with prior work, and we use it without modification. The MATH-500 parser is our own. For each run, we inspect twenty randomly selected extractions and report the fraction of responses for which the parser finds a strategy.

This leaves an important limitation. Without a larger human audit, we cannot measure precisely how often the MATH-500 parser merges two different opening calculations or splits one underlying approach into multiple families.

A.4 Inclusion thresholds and coverage

We keep a strategy family only if it appears at least 5 times in the base model or, for Countdown, at step 25. A problem is included only if it has at least 2 such families. This prevents quality μ from being estimated almost entirely from one or two samples. The smoothing in Equation 2 further reduces instability from small counts.

Table 3 summarizes MATH-500 coverage. For Countdown, the fixed family set defined at step 25 contains $3{,}218$ qualifying families across 969 of the 1000 sampled problems.

Coverage is similar across the three model pairs. The parser assigns a family to about three quarters of responses. After the minimum-count rule, the analyses contain 378–485 strategy families across 78–88 problems.

Figure 5 summarizes the analysis. Responses are grouped by their opening calculation, each family is described by its frequency q and quality μ, and the selection models predict how family frequencies change. The last panel shows the separate finite-sampling simulation. In that simulation, a family that receives zero of G draws is removed under an explicit modeling assumption. This simulated removal is distinct from empirical disappearance, which means zero counts in the 64-sample Countdown evaluation.

B Estimation Details

B.1 Fitting the selection models

Each selection model predicts a probability distribution over the strategy families for one problem. We choose model parameters that make the observed later counts as likely as possible.

Let n_{xb} be the number of later responses to problem x assigned to family b, and let $p_{xb}(\theta)$ be the probability assigned by a model with parameters θ. We maximize

$$\ell(\theta) = \sum_{x} \sum_{b \in \mathcal{B}(x)} n_{xb} \log p_{xb}(\theta). \tag{10}$$

This is a multinomial log-likelihood. Larger values mean that the model assigns more probability to the counts we actually observe.

For numerical stability, probabilities are normalized in log space. For Model C, the frequency-plus-quality model,

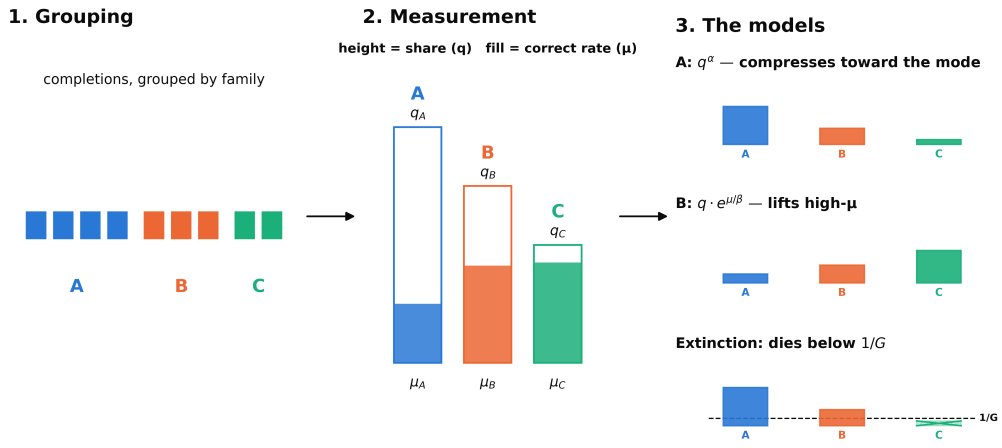

Figure 5: Method overview. Sampled responses are grouped into strategy families by their opening computation. Each family is summarized by frequency q and quality μ. Candidate reweighting rules predict later family frequencies. The last panel shows the separate finite-sampling simulation, where zero draws in a group of size G cause simulated removal.

$$\log p_{xb} = \alpha_C \log\left(q_0(b) + \epsilon\right) + \frac{\mu_0(b)}{\beta_C} - \log Z_x. \tag{11}$$

Model A uses only the first term and fits α_A. Model B fixes the frequency exponent at 1 and fits only $1/\beta_B$. Model C fits both α_C and $1/\beta_C$. This notation matches Models A, B, and C in Equations 3–5.

We optimize with L-BFGS-B under

$$0.01 \le \alpha \le 20, \qquad -20 \le \frac{1}{\beta} \le 20. \tag{12}$$

No fitted parameter approaches either boundary.

For Countdown, q_0 is the step-25 distribution and every later checkpoint is fit relative to it. The resulting α_C values are cumulative. They measure how strongly a later distribution reflects the initial frequency imbalance. They are not estimates of a per-update exponent. Families not observed at a later checkpoint remain in the fixed family set with count zero.

We compare models using AIC, BIC, likelihood-ratio tests for nested models, and five-fold cross-validation. Cross-validation holds out whole problems, so families from one problem never appear in both training and validation folds.

B.2 Within-problem standardization

Strategies from different problems can have very different baseline frequencies and qualities. For the strategy-level regressions, we therefore compare each family with the other families available for the same problem. For a predictor z and family b from problem x, we compute

$$\tilde{z}(b) = \frac{z(b) - \bar{z}(x)}{s_z(x)}, \tag{13}$$

$$\bar{z}(x) = \frac{1}{n_x} \sum_{b' \in \mathcal{B}(x)} z(b'), \tag{14}$$

$$s_z(x) = \sqrt{\frac{1}{n_x - 1} \sum_{b' \in \mathcal{B}(x)} (z(b') - \bar{z}(x))^2}. \tag{15}$$

Here, $n_x = |\mathcal{B}(x)|$. We standardize $\log q_t$ and μ_t separately. A positive standardized frequency means more common than the other strategies for that problem, not common in an absolute sense. If a predictor has no within-problem variation, we set its standardized value to zero.

B.3 Observed-disappearance regression

For a Countdown transition, define $D_t(b) = 1$ when family b is observed at checkpoint t but receives zero counts among the 64 evaluation samples at checkpoint $t + 1$. Otherwise $D_t(b) = 0$. This event is empirical non-observation at sample size 64. It does not imply that the true policy probability is zero.

Among families observed at checkpoint t, we fit

$$\Pr[D_t(b) = 1 \mid q_t(b) > 0] = \sigma\Big(\gamma_0 + \gamma_q \widetilde{\log q}_t(b) + \gamma_\mu \tilde{\mu}_t(b)\Big), \qquad \sigma(u) = \frac{1}{1 + e^{-u}}. \tag{16}$$

The coefficient γ_q measures the association between current frequency and next-checkpoint disappearance while holding quality fixed. The coefficient γ_μ does the same for quality. We use an effectively unregularized fit with $C = 10^9$.

B.4 Change-in-frequency regression

We also study continuous changes in frequency. For a target checkpoint t, we define

$$\delta_t(b) = \log(q_t(b) + \epsilon) - \log(q_0(b) + \epsilon), \qquad \epsilon = 1/512. \tag{17}$$

The constant ϵ is only a numerical floor that allows the logarithm to be evaluated when a family has zero observed count. We choose $1/512$ because it is smaller than one count in the 64-sample Countdown evaluation and also smaller than one count in the 256-sample MATH-500 evaluation. It should not be interpreted as a measured probability. Results are driven by the ordering of common and rare families rather than by treating ϵ as an observed frequency.

For Countdown, we fit

$$\delta_t(b) = \eta_0 + \eta_q \widetilde{\log q}_0(b) + \eta_\mu \tilde{\mu}_0(b) + \varepsilon(b). \tag{18}$$

The main trajectory table uses the fixed step-25 family set. At late checkpoints, this linear regression is affected by strong floor and ceiling constraints. Many rare families already sit at the zero-count floor and cannot show another large negative change, while dominant families are close to probability one and have little room to increase. For this reason, a late negative η_q should not be read as evidence that the cumulative frequency advantage reverses. The multinomial selection model is the primary analysis of the full distribution.

We also use a separate MATH-500 split-sample diagnostic in Section C to estimate how much sampling noise limits a strategy-level change regression. That diagnostic uses 256 responses per problem and is not part of the reported Countdown trajectory coefficients.

B.5 CALIBRATION OF PREDICTED CONCENTRATION

We test whether Model A predicts the amount of diversity loss observed across problems. For a target checkpoint t, the smooth model predicts

$$\hat{q}_t(b) = \frac{\exp(\alpha_{A,t} \log q_0(b))}{\sum_{b'} \exp(\alpha_{A,t} \log q_0(b'))}, \qquad \hat{N}_{\mathrm{eff},t}(x) = \exp(H(\hat{q}_t)) \,. \tag{19}$$

We compare predicted and observed effective strategy counts with

$$N_{\mathrm{eff},t}(x) = c_0 + c_1 \hat{N}_{\mathrm{eff},t}(x) + \varepsilon(x). \tag{20}$$

We call c_1 the calibration slope. A value near 1 means that predicted variation across problems has about the observed scale. A much smaller value means the model underpredicts concentration.

We then add a simplified finite-sampling process. For each G and each problem, we run 1000 simulations. The reference probabilities remain the step-25 frequencies q_0. At each later Countdown checkpoint, we use that checkpoint's fitted $\alpha_{A,t}$, restrict to families not previously removed in the simulation, and compute

$$q_{\mathrm{sim},t} = \mathrm{softmax}(\alpha_{A,t} \log(q_0 + \epsilon)) \,. \tag{21}$$

We draw G samples from this distribution. A family receiving zero draws is removed for all later simulated checkpoints, and the remaining probabilities are renormalized. Thus, the state carried across simulated checkpoints is the surviving-family set, while q_0 remains the reference distribution and the cumulative exponent changes with checkpoint. We apply one pruning stage for each later checkpoint 50, 100, 200, 300, and 450.

Permanent removal after zero draws is deliberately stronger than what one real GRPO update guarantees. In real training, an unsampled family receives no direct sample in that group, but shared parameters can still change its probability. The simulation therefore tests whether finite groups can amplify concentration in the observed direction, not whether this is the complete training mechanism.

We evaluate $G \in \{8, 16, 32, 64\}$. The released Countdown configuration uses $G = 8$.

B.6 UNCERTAINTY

All reported confidence intervals are 95% bootstrap intervals based on 1000 resamples. We resample whole problems rather than individual responses, so each sampled problem brings all of its families and responses with it. We use `np.random.default_rng(20260913)`. Given the cached generations, all bootstrap results are deterministic.

C VALIDATION

The cumulative α trend is not an estimator artifact. Because the observed distributions become more concentrated, one concern is that the fitted cumulative α could drift upward mechanically even when the generating exponent is fixed. We test this with synthetic data generated at a known constant α. The simulation matches the sample count and frequency structure used by the estimator.

Table 4 shows that the estimator recovers the generating value closely and does not drift upward on its own. This rules out a simple estimation artifact. It does not show that the per-update selection strength in the real training run increases, because the real checkpoint fits are cumulative relative to step 25.

Running the same synthetic check after restricting to families that remain observed, or to problems that keep several observed families, produces a downward bias. This motivates keeping the original step-25 family set in the main cumulative analysis.

Table 4: Recovered α from synthetic data generated with a constant true value. The fixed-family-set estimator is used at every checkpoint. This check tests estimator drift, not whether real per-update selection strength changes.

True α	Step 50	Step 100	Step 200	Step 300	Step 450
1.0	1.000	0.993	0.999	0.991	0.992
1.5	1.503	1.519	1.490	1.486	1.504
2.0	1.988	2.002	1.996	1.995	2.000
3.0	2.964	3.005	3.013	3.001	3.008

Table 5: Model C frequency parameter under three family-selection rules. The fixed set retains later-zero families. The restricted set uses only families observed at the target checkpoint. The panel uses only the 224 resistant problems.

Step	Fixed set (α_C)	Restricted (α_C)	Panel (α_C)
50	1.404	1.343	1.217
100	1.892	1.639	1.203
200	2.814	1.506	0.849
300	3.082	0.964	0.871
450	3.009	0.929	0.728

Conditioning on observed survivors changes the question. Table 5 compares three family-selection rules. The fixed analysis retains all step-25 families, including those with later count zero. The restricted analysis keeps only families observed at the target checkpoint. The panel analysis keeps only the 224 resistant problems.

These fits answer different questions. The fixed-set estimate describes how the full initial family distribution maps to a later distribution. The restricted fit describes only families still observed later, after the largest losses have already occurred. A lower restricted α_C therefore does not imply that the overall path dependence reverses.

Table 6 shows how little room remains among observed families late in training.

By step 200, the median problem places 93.8% of its observed mass on one family. By step 300, the median is 100%. This strong floor-and-ceiling structure also explains why the late linear change-regression coefficient can flip sign even though the cumulative multinomial frequency effect remains large.

Quality estimates contain repeatable signal. A weak quality coefficient could arise simply because μ is measured poorly. Table 7 reports the spread of μ and a reliability measure. Here reliability asks whether the ordering of family-quality estimates would be similar if the available responses were measured again. Where two independent halves are available, we use split-half correlation with the usual Spearman-Brown adjustment. For settings where that split is too sparse, we report the stated analytical lower bound from binomial sampling.

The values are high enough that quality is not pure sampling noise. MATH-500 also has substantially more between-family quality variation than Countdown. This makes the small or unstable strategy-level quality coefficients harder to explain solely by a lack of measurable variation.

As an additional MATH-500 diagnostic, we restrict to 134 families with at least 20 base-model observations across 71 problems. In that high-count subset, the standardized frequency coefficient is 0.066 $[0.011, 0.129]$, while the quality coefficient is 0.011 $[-0.060, 0.078]$. This diagnostic is separate from the Countdown trajectory regression. It supports the narrower claim that frequency is the more stable strategy-level signal.

Quality helps the distribution-level model more than the linear change regression. The positive $1/\beta_C$ values in the multinomial models and the small η_μ values in some change regressions are not estimates of the same quantity. Model C fits the whole normalized family distribution

Table 6: Percentiles of the largest observed family share within each Countdown problem, $\max_b q_t(b)$.

Step	p10	p25	p50	p75	p90
50	0.266	0.344	0.438	0.531	0.594
100	0.375	0.500	0.625	0.703	0.766
200	0.562	0.812	0.938	0.984	1.000
300	0.641	0.938	1.000	1.000	1.000
450	0.672	0.953	1.000	1.000	1.000

Table 7: Variation and reliability of strategy quality μ. Reliability measures repeatability of the estimated family-quality differences under resampling.

Setting	$\mathrm{Var}(\mu)$	Within-problem SD	Reliability
MATH-500 (pooled)	0.0768	0.158	0.940
Countdown	0.0137	0.058	0.830
Math-7B (specialized)	0.0768	0.141	0.767
Qwen2.5-1.5B (general)	0.0891	0.111	0.797
Qwen2.5-7B (general)	0.0828	0.115	0.796

jointly, so a quality term can improve likelihood even when its marginal linear association with noisy log-frequency changes is small. The change regression instead asks whether within-problem standardized quality predicts a family's measured δ_t after a logarithm, a zero-count floor, and strong late-stage ceilings. We therefore use the multinomial comparison to say that quality adds some predictive information, while using the regression and disappearance results to avoid claiming a large, stable independent quality effect.

Sampling noise limits the MATH-500 change regression. The MATH-500 endpoint analysis has 256 responses per problem, which allows a split-sample noise check. We divide those responses into two halves and compute a family's frequency change separately in each half. The two estimates have correlation 0.279. After the Spearman-Brown adjustment, the estimated reliability is 0.436.

The MATH-500 change regression reaches $R^2 = 0.072$ when predictors and outcomes come from the same samples and $R^2 = 0.112$ when they come from independent halves. Relative to the estimated noise ceiling, these are 16.5% and 25.7% of explainable variation. In the independent-half analysis, the standardized frequency coefficient is 0.262 and the quality coefficient is 0.001. These values are reported only as a noise diagnostic and are not part of the Countdown trajectory table.

Late Countdown concentration is not duplicate text generation. A final concern is that the apparent strategy collapse might simply reflect the model repeating nearly identical responses. We test this on 50 Countdown problems. Mean pairwise semantic similarity is 0.800 (SD 0.011) at step 25 and 0.777 (SD 0.135) at step 450. The fraction of distinct generated traces at step 450 has mean 0.915, median 1.00, and minimum 0.44 on the most concentrated problem.

Late checkpoints can therefore produce many textually different responses while repeatedly choosing the same opening strategy. The observed collapse is not simply duplicate generation.

D Additional Experimental Details

D.1 Models

Table 8 lists the public model repositories and revisions used in our experiments. We did not train any of these models ourselves.

Table 8: Model pairs and revisions used in the experiments.

Setting	Role	Repository and revision
Math-7B (specialized)	base	`Qwen/Qwen2.5-Math-7B` `b101308f`
	RL	`hkust-nlp/Qwen-2.5-Math-7B-SimpleRL-Zoo` `c9e929d4`
Qwen2.5-1.5B (general)	base	`Qwen/Qwen2.5-1.5B` `8faed761`
	RL	`hkust-nlp/Qwen-2.5-1.5B-SimpleRL-Zoo` `c014edac`
Qwen2.5-7B (general)	base	`Qwen/Qwen2.5-7B` `d1497293`
	RL	`hkust-nlp/Qwen-2.5-7B-SimpleRL-Zoo` `d630142f`
Countdown trajectory	–	`philschmid/qwen-2.5-3b-r1-countdown` `8da99c42`, base architecture Qwen2.5-3B-Instruct, GRPO, checkpoints 25, 50, 100, 200, 300, 450

The three MATH-500 comparisons cover a math-specialized 7B model and general-purpose models at 1.5B and 7B parameters. Countdown provides six checkpoints from one GRPO run, which is why it supports trajectory-level claims while MATH-500 serves as endpoint replication.

D.2 Sampling

For MATH-500, we draw 100 problems once from the `HuggingFaceH4/MATH-500` test split and reuse that set for all models. For each problem and model, we sample 256 responses using temperature 1.0, top-p 1.0, and maximum length 2048. The prompt contains the raw problem text with no system prompt.

Per-problem random seeds are generated deterministically as

$$\mathrm{int}(\mathrm{sha256}(\texttt{SEED}\,|\,\texttt{model}\,|\,\texttt{problem_id}\,|\,\texttt{generation})[:4])\,, \tag{22}$$

with `SEED = 20260913`.

For Countdown, we use 1000 problems from the training split of `Jiayi-Pan/Countdown-Tasks-3to4`. We sample 64 responses per problem at each checkpoint with temperature 1.0, top-p 1.0, maximum length 1024, and bfloat16. Prompts use the checkpoint's native system prompt and an assistant prefill opening a `<think>` block.

All generations use vLLM with `enforce_eager=True` and `enable_prefix_caching=False`. The maximum model length is 4096 for the replication and Countdown runs and is left unset for the specialized 7B run.

D.3 Correctness

For Countdown, correctness is determined by the exact solver in Appendix A.

For MATH-500, we extract the final `\boxed{}` answer and normalize both the model answer and reference. Normalization removes presentational differences such as math delimiters, display commands, spacing, thousands separators, and trailing periods. It also standardizes fraction commands and common text-formatting commands.

We first compare normalized answers as strings. If that is inconclusive, we use symbolic checks in SymPy. We try direct rational conversion, symbolic parsing, and a LaTeX-to-Python conversion, followed by

Table 9: Inference compute. Hardware was assigned by the scheduler under a compute-capability ≥ 8.0 constraint, so the exact GPU model was not recorded for every run.

Setting	GPU-hours	Hardware
Math-7B (specialized)	3.97	1 GPU, CC ≥ 8.0
Qwen2.5-1.5B (general)	1.81	1 GPU, CC ≥ 8.0, 64 GB RAM
Qwen2.5-7B (general)	4.68	1 GPU, CC ≥ 8.0, 64 GB RAM
Countdown (6 checkpoints)	7.40	1 $\times$ A5000, 48 GB RAM
Total	17.86	

```
sympy.simplify(a-b) == 0.
```
(23)

This allows mathematically equivalent answers with different surface forms to count as correct.

D.4 Rollout group size

The finite-sampling analysis uses G, the number of responses generated for one prompt in a training update.

For Countdown, the released configuration sets `num_generations: 8`. Training used TRL 0.14.0 with `use_vllm: true` and three training processes. Prompts are distributed across processes, while vLLM still generates 8 completions for each prompt. The group size is therefore $G = 8$, not 8×3.

For SimpleRL-Zoo, the training script sets `ROLLOUT_N=8`, passed to the trainer as `actor_rollout_ref.rollout.n=8`. The accompanying paper also reports 8 responses per prompt. The released weights do not contain this metadata, so we take the value from the associated training configuration.

Both training setups therefore use $G = 8$.

D.5 Compute

Table 9 reports inference compute. No models were trained. The analysis scripts run on CPU in seconds to minutes and were not timed separately.

The complete evaluation uses 17.86 GPU-hours. Countdown accounts for 7.40 hours because generations are collected at six checkpoints, while each MATH-500 setting requires only a base/RL endpoint comparison.

E Additional Results

E.1 Full trajectory parameters

Table 10 gives the values behind the Countdown trajectory plots. Every α_C and $1/\beta_C$ value is from Model C and is fit cumulatively relative to step 25. The disappearance column is different. It is a one-transition rate among families observed at the preceding checkpoint.

The cumulative frequency parameter grows from 1.404 at step 50 to about 3 by step 300, while effective diversity falls from 2.909 to 1.152. The largest single-transition disappearance rate is 41.2% from step 100 to step 200. Because the parameter fits are cumulative and the disappearance rate is local to one transition, these columns should not be interpreted as the same kind of time-varying quantity.

E.2 Change in frequency across the trajectory

Table 11 reports the Countdown linear change regression. The regression uses the fixed family set and the log floor defined in Appendix B.

Table 10: Countdown trajectory on the fixed step-25 family set. Model C parameters are cumulative relative to step 25. Observed disappearance is the fraction present at the preceding checkpoint and absent from all 64 samples at the listed checkpoint.

Step	α_C [95% CI]	$1/\beta_C$ [95% CI]	Disappearance	N_{eff}
25	–	–	–	2.909
50	1.404 [1.371, 1.437]	0.693 [0.426, 0.949]	2.6%	2.435
100	1.892 [1.828, 1.959]	1.759 [1.218, 2.272]	10.6%	1.924
200	2.814 [2.641, 3.002]	3.106 [1.795, 4.467]	41.2%	1.315
300	3.082 [2.859, 3.356]	3.576 [1.964, 5.192]	25.7%	1.160
450	3.009 [2.775, 3.287]	3.647 [2.011, 5.224]	5.5%	1.152

Table 11: Change-in-frequency regression on Countdown. Predictors are standardized within each problem. Late coefficients are affected by zero-count floors and probability ceilings and are not used as the primary selection estimate.

Step	η_q [95% CI]	η_μ [95% CI]	R^2
50	0.049 [0.027, 0.071]	0.037 [0.015, 0.060]	0.010
100	0.085 [0.034, 0.131]	0.051 [0.003, 0.095]	0.011
200	−0.101 [−0.253, 0.065]	−0.014 [−0.151, 0.145]	0.007
300	−0.238 [−0.459, −0.030]	−0.005 [−0.222, 0.198]	0.040
450	−0.342 [−0.579, −0.083]	0.071 [−0.152, 0.315]	0.096

Frequency predicts positive log-frequency change early. Later, the coefficient turns negative. With the fixed family set, that sign change cannot be explained by dropping disappeared families from the regression. Instead, the outcome is strongly bounded late in training. Many rare families already sit at the zero-count floor, while dominant families sit near the probability-one ceiling. This makes a linear change coefficient difficult to interpret after collapse. The cumulative multinomial fits and the observed-disappearance regressions are therefore the primary evidence for the frequency effect.

E.3 Which problems resist concentration

We predict whether a Countdown problem is resistant using only step-25 features. A resistant problem has at least two step-25 qualifying families that remain observed, with at least one of 64 evaluation samples, at every later checkpoint.

"Observed qualifying families" is the raw number of step-25 families that meet the five-sample threshold. "Entropy-effective family count" is $\exp(H)$ computed from their step-25 frequency distribution. "Solver-valid entrance families" counts first-step entrances for which the exact solver can construct at least one correct Countdown solution.

Initial entropy is the clearest predictor. Higher entropy means probability is spread more evenly, and these problems are more likely to retain multiple observed strategies. The raw number of families has a negative coefficient once this evenness is controlled for. Having many families is therefore not enough if most of the probability already lies on one of them.

None of the quality-related variables has a confidence interval clearly away from zero. Resistant problems begin somewhat harder, with step-25 accuracy 6.98% rather than 9.19%, a difference of 2.22 points $[0.79, 3.52]$. By step 450, the two groups reach 42.17% and 45.67%. The final 3.50-point difference has interval $[-2.16, 9.84]$.

E.4 Calibration under finite sampling

Table 13 compares the smooth Model A prediction with the simplified finite-sampling simulation.

Adding the finite-sampling rule improves calibration at every checkpoint, and the largest improvement occurs for $G = 8$. At step 50 the slope rises from 0.780 to 1.005, and at step 200 it rises from

Table 12: Predicting resistance from step-25 features. There are 969 analysis problems, of which 224 are resistant. Overall AUC is 0.702 $[0.679, 0.733]$.

Feature	Coefficient	95% CI
Entropy	1.854	[1.088, 2.659]
Observed qualifying families	−1.547	[−2.616, −0.519]
Entropy-effective family count	0.415	[−0.611, 1.465]
Maximum family share	0.089	[−0.281, 0.445]
Mean quality	0.072	[−0.447, 0.551]
Variance of quality	0.126	[−0.028, 0.270]
Base accuracy	−0.249	[−0.758, 0.251]
Solver-valid entrance families	0.085	[−0.027, 0.199]

Table 13: Calibration slope of observed effective strategies against predicted effective strategies. The smooth model uses cumulative Model A reweighting. The augmented models also apply the simulated zero-draw removal rule. $G = 8$ is the released training group size.

Model	Step 50	Step 100	Step 200	Step 300	Step 450
Smooth	0.780	0.506	0.180	0.067	0.058
Augmented, $G = 8$	**1.005**	**0.699**	**0.288**	**0.125**	**0.119**
Augmented, $G = 16$	0.839	0.551	0.210	0.091	0.083
Augmented, $G = 32$	0.789	0.509	0.184	0.078	0.069
Augmented, $G = 64$	0.782	0.506	0.180	0.076	0.067

0.180 to 0.288. The $G = 8$ model still reaches only 0.119 at step 450, so this simplified mechanism accounts for only part of the late collapse.

Table 14 describes empirical non-observation. The absent counts are snapshots, not cumulative totals. A family can be absent at one checkpoint and later reappear.

The fraction of families below the rough $1/G$ scale rises from 0.291 at step 50 to more than one half by step 200. Non-observation also becomes more persistent across the sampled checkpoints. This pattern is consistent with a role for limited sampling, but it does not establish that those families have exactly zero policy probability.

E.5 Model comparison

Table 15 compares Models A, B, and C on MATH-500.

Model C has the lowest NLL in all three settings. Neither single-factor model wins everywhere. Model A fits better for the specialized 7B and general 1.5B models, while Model B fits better for the general 7B model.

For the specialized 7B pair, AIC is 28353.0, 28391.1, and 28334.2 for Models A, B, and C. BIC is 28360.3, 28398.5, and 28348.8. Five-fold cross-validated log-likelihood is -14179.1, -14194.9, and -14169.6. These criteria also favor Model C. This is why we say that quality adds distribution-level predictive information even though the simpler strategy-level quality coefficients are weaker and less stable.

E.6 Where the mathematical-reasoning endpoint sits

For a rough cross-setting comparison, Table 16 reports only the frequency-only Model A parameter on families that remain observed in both members of the comparison. This survivor-only α_A is deliberately not the same quantity as the fixed-set α_C in Table 5. Removing the duplicate disappearance and diversity columns also avoids mixing numbers computed under different family definitions.

The specialized MATH-500 endpoint is closest to early Countdown under this survivor-only frequency fit. We do not use the later decline in this table as evidence that overall frequency selec-

Table 14: Sampling vulnerability and persistence for $G = 8$. "Families absent" means zero counts in the 64-sample evaluation at that checkpoint. "Still absent afterward" counts those that never reappear in any later sampled checkpoint.

	Step 50	Step 100	Step 200	Step 300	Step 450
Fraction below $1/G$	0.291	0.414	0.514	0.535	0.528
Families absent	85	381	1534	1954	1964
Still absent afterward	48	365	1509	1895	1964
Fraction still absent	0.565	0.958	0.984	0.970	1.000

Table 15: Negative log-likelihood (NLL), where lower is better. Model A uses frequency only, Model B quality only, and Model C both.

Setting	NLL (A)	NLL (B)	NLL (C)
Math-7B (specialized)	14175.5	14194.6	**14165.1**
Qwen2.5-1.5B (general)	11274.8	11369.8	**11208.8**
Qwen2.5-7B (general)	14701.7	14675.9	**14645.7**

tion reverses, because conditioning on survival removes exactly the families that experienced the strongest losses.

E.7 Diversity loss across settings

Table 17 compares the effective number of strategies before and after RLVR. Countdown uses the fixed step-25 family set and spans the full trajectory. The other rows are base/RL endpoint comparisons.

All four settings lose strategy diversity, but the size differs sharply. The MATH-500 pairs lose 12.3–18.7%, while Countdown loses 60.4% over the full trajectory and ends at 1.152 effective strategies per problem.

The cross-setting evidence therefore supports the broad frequency-selection result more strongly than the late-stage persistent-disappearance result.

Table 16: Survivor-only Model A frequency parameter for the specialized MATH-500 endpoint and Countdown checkpoints. These values are for rough placement only and are not the fixed-set cumulative estimates used in the main analysis.

Setting	Survivor-only α_A
MATH-500 (specialized 7B)	1.112
Countdown, step 50	1.065
Countdown, step 100	1.098
Countdown, step 200	0.764
Countdown, step 300	0.369
Countdown, step 450	0.215

Table 17: Effective number of strategies before and after RLVR.

Setting	Before	After	Reduction
Countdown (trajectory)	2.909	1.152	60.4%
Math-7B (specialized)	4.243	3.723	12.3%
Qwen2.5-1.5B (general)	3.736	3.038	18.7%
Qwen2.5-7B (general)	4.104	3.587	12.6%

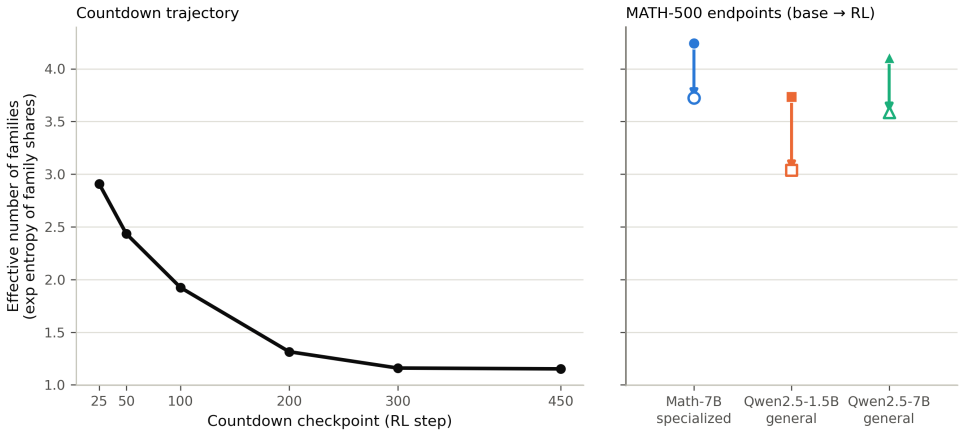

Figure 6: RLVR reduces strategy diversity in all four settings, with a much larger reduction over the full Countdown trajectory than in the MATH-500 endpoint pairs.

E.8 RARE BUT RELIABLE STRATEGIES THAT BECOME UNOBSERVED

The regressions show that high quality does not consistently protect a rare strategy from later non-observation. Table 18 gives five examples. Counts and frequencies are measured at step 25. "Became unobserved" gives the first transition in which the family receives zero of 64 evaluation samples.

For readability, we report both raw success counts and smoothed quality μ. The smoothing adds one hypothetical success and one hypothetical failure. For the $61-$ family, 5 of 6 observed attempts are correct, giving smoothed $\mu = 0.750$.

Each example has the same descriptive pattern. The family that later becomes unobserved has higher measured quality but lower step-25 frequency than a family that remains observed on the same problem. For $\{94, 80, 85\} \to 89$, the $94+$ family succeeds on $3/5$ attempts and has smoothed quality 0.571, but its frequency is only 0.078. The $94-$ family succeeds on $10/32$ attempts with smoothed quality 0.324, yet starts at frequency 0.500.

These examples do not establish causality. They illustrate the same pattern as the disappearance regressions. A strategy can be relatively reliable and still fall below detection after becoming rare.

Table 18: Rare, higher-quality entrance families that later become unobserved, paired with a lower-quality family that remains observed on the same problem.

		Later-unobserved family			Observed family		
Problem	Became unobserved	Entrance	Correct	μ / q	Entrance	Correct	μ / q
$\{61, 47, 68\} \to 82$	$100 \to 200$	$61-$	5/6	0.750 / 0.094	$68-$	6/20	0.318 / 0.312
$\{15, 19, 12\} \to 22$	$200 \to 300$	$15+$	4/6	0.625 / 0.094	$19-$	9/20	0.455 / 0.312
$\{32, 12, 66\} \to 86$	$100 \to 200$	$86-$	5/8	0.600 / 0.125	$32+$	4/10	0.417 / 0.156
$\{94, 80, 85\} \to 89$	$100 \to 200$	$94+$	3/5	0.571 / 0.078	$94-$	10/32	0.324 / 0.500
$\{80, 96, 79\} \to 97$	$200 \to 300$	$80+$	3/5	0.571 / 0.078	$96-$	10/29	0.355 / 0.453