A Closer Look at RLVR Through the Lens of Reasoning Strategies

Anonymous ACL submission

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) improves reasoning models, but typically collapses the space of reasoning approaches they employ. As test-time search methods increasingly rely on diverse candidate pools, understanding and quantifying this narrowing becomes critical. Standard metrics offer limited insight: pass@k measures aggregate accuracy but cannot distinguish why two methods differ, and token entropy shows no consistent relationship with either single-sample accuracy or repeated-sampling gains. To fill this gap, we propose STRATEGY-F1, which evaluates a policy by its reasoning diversity along two axes: COVERAGE(retention of the base model's approach repertoire) and SHARPNESS(concentration on the most viable approach per problem). The coverage-sharpness decomposition captures the tradeoff consistently: methods with higher F1 exhibit a healthy balance between single-sample performance and gains realized with repeated sampling.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become the dominant paradigm for improving reasoning in language models (Shao et al., 2024; Yu et al., 2026). These methods reliably improve single-sample accuracy (pass@1), but how they reshape the model's repertoire of reasoning approaches has received limited attention. This matters increasingly as inference-time techniques (Wu et al., 2024; Zhang et al., 2025), from best-of-n selection (Snell et al., 2024) to evolutionary methods, becomes standard practice. These methods rely on diverse candidates, and if RL training collapses the model onto a narrow set of approaches, additional samples become redundant and test-time compute yields diminishing returns.

Recent work has shown that RLVR does not enable models to discover new problem-solving methods beyond those available to the base model (Yue et al., 2017). This leaves open a set of critical questions: among the methods the base model already has, how many survive training, how many are pruned, what are the implications, and how to *quantify* this?

In this paper, we study these questions across 6 different RLVR algorithms: GRPO (Shao et al., 2024), DAPO (Yu et al., 2026), R-Ada (Xiong et al., 2025), GRESO (Zheng et al., 2026), and KnapsackRL (Li et al., 2025). We analyze how these methods affect retention and strengthening of diverse reasoning strategies and the implication on final performance of the policy. Our analysis yields two main contributions:

1. **Token entropy does not capture the coverage–sharpness interaction.** We find that token entropy, the standard diversity diagnostic in language models (Cui et al., 2025; Yu et al., 2026), shows no consistent relationship with single-sample accuracy or repeated sampling performance, across five methods (Section 3). This motivates the need for a reasoning-level metric.

2. **We propose STRATEGY-F1**, which decomposes reasoning diversity into *coverage*, how many of the base model's approaches a trained policy retains and *sharpness* how effectively it concentrates on approaches that solve each problem. Our metric captures the balance between focus and breadth that directly determine pass@1 and how much a method benefits from repeated sampling, respectively. Across five methods and two benchmarks, F1 tracks the pass@1 vs. pass@k tradeoff that token entropy cannot (Table 2).

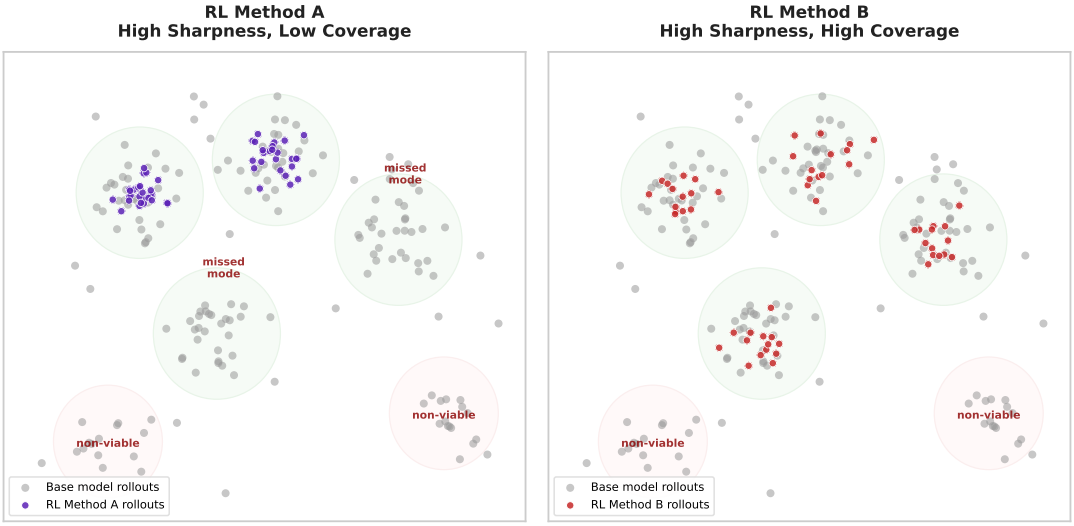

Figure 1: Conceptual illustration of coverage and sharpness. Grey dots: base model rollouts across reasoning strategies (clusters). Green regions: viable strategies; pink: non-viable. **Left:** RL Method A concentrates on few strategies (high sharpness) but abandons others (low coverage). **Right:** Method B covers more viable strategies while maintaining concentration within each (high sharpness, high coverage). Strategy F1 rewards Method B for balancing both axes.

2 Related Work

Diversity in RLVR. Token-level entropy is the standard diversity diagnostic (Cui et al., 2025; Yu et al., 2026), but does not capture strategy-level behavior. Yue et al. (2017) establish that RLVR does not expand the set of solvable problems beyond the base model; our coverage metric provides a finer view by measuring which reasoning approaches are retained or lost. A growing line of work proposes training-time interventions to preserve diversity, including diversity-aware rewards (Chen et al., 2025), frequency-aware sampling (Anschel et al., 2025), risk-sensitive objectives (Jiang et al., 2025), and exploration-driven optimization (Li et al., 2026). These methods aim to *boost* diversity, we contribute a metric to *measure* the coverage-sharpness tradeoff they navigate.

Model	entropy	pass@1	pass@64
Base	.191	18.2%	56.7%
GRPO	.158	29.2%	63.3%
DAPO	.274	28.9%	66.7%
R-Ada	.376	25.9%	70.0%
GRESO	.140	28.7%	60.0%
KnapsackRL	.318	29.9%	56.7%

Table 1: Token entropy (AIME-2024) shows no consistent trend with pass@1 or pass@64 across six policies.

Test-time scaling and distributional tradeoffs. Inference-time methods increasingly rely on diverse candidate pools (Zhang et al., 2025; Snell et al., 2024; Yao et al., 2023), and Bahlous-Boldi et al. (2026) argue that RL post-training should maximize solution diversity for downstream search. Our decomposition mirrors the precision–recall framing of Cha and Cho (2026) for generative models: coverage corresponds to recall over reasoning approaches, sharpness to precision in per-problem strategy selection.

3 Preliminary: Token Entropy Does Not Track Reasoning Diversity

The implicit assumption is that higher entropy indicates broader exploration of the solution space.

Table 1 empirically finds this relationship to be inconsistent. Across five methods trained from the same base model, token entropy varies in a broad range (0.140 to 0.376), yet neither to pass@1 or pass@64 follows this ordering. If entropy tracked reasoning diversity, one would expect methods with higher entropy to achieve higher pass@64, since a more diverse policy should cover more solution paths under repeated sampling. Instead:

- GRESO records a very low entropy (0.140) but maintains a higher pass@64 than the base policy or KnapsackRL both of which record higher entropies.
- R-Ada has the highest entropy (.376) and the highest pass@64 (70.0%), consistent with the

assumption - but KnapsackRL, with comparably high entropy (.318), achieves only 56.7% pass@64, the lowest recorded in our experiments, matching the base policy.

These inconsistencies indicate that token entropy captures surface-level diversity behaviors as opposed to diversity relevant to genuinely diverse reasoning paths. A method can exhibit high entropy through lexical variation across rollouts while converging on the same underlying reasoning approach, or conversely, exhibit low entropy through formulaic phrasing while alternating between distinct solution methods.

4 Method

Standard pass@k measures aggregate correctness but cannot distinguish *how* two RLVR methods differ. We decompose reasoning diversity into COVERAGE (retention of the base model's strategy repertoire) and SHARPNESS (concentration on viable strategies per problem).

Strategy profiling. For each model π and problem p, we sample N solutions (fixed across all models and problems). An LLM judge labels each solution with a strategy $s_j \in \mathcal{S} = \{s_1, \ldots, s_K\}$ and we verify correctness $y_j \in \{0, 1\}$ against the known answer. From these samples we estimate the strategy distribution and per-strategy correctness rate:

$$\hat{\pi}(s \mid p) = \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}[s_j = s], \tag{1}$$

$$\hat{c}_\pi(s, p) = \frac{\sum_j \mathbf{1}[s_j = s,\ y_j = 1]}{\sum_j \mathbf{1}[s_j = s]}, \tag{2}$$

with $\hat{c}_\pi(s, p) = 0$ when s is never sampled. A strategy s is *viable* for problem p if either the base model or the model under evaluation produces at least one correct solution using it:

$$\mathcal{V}(p) = \{s \in \mathcal{S} : \hat{c}_{\pi_{\text{base}}}(s, p) > 0 \text{ or } \hat{c}_{\pi_{\text{rl}}}(s, p) > 0\}. \tag{3}$$

This makes all downstream metrics properties of the (base, RLVR method) pair alone. Problems where $\mathcal{V}(p) = \emptyset$ are excluded.

Coverage. Let $\mathcal{S}_\pi = \{s \in \mathcal{S} : \exists p,\ \hat{\pi}(s \mid p) > 0\}$ be the strategies observed from model π across all problems. COVERAGE measures what fraction of the base model's repertoire survives RLVR:

$$\text{Cov}(\pi_{\text{rl}}) = \frac{|\mathcal{S}_{\pi_{\text{rl}}} \cap \mathcal{S}_{\pi_{\text{base}}}|}{|\mathcal{S}_{\pi_{\text{base}}}|}. \tag{4}$$

Sharpness. SHARPNESS measures how concentrated the model is on its best viable strategy:

$$\sigma(\pi_{\text{rl}}, p) = \max_{s \in \mathcal{V}(p)} \hat{\pi}_{\text{rl}}(s \mid p), \tag{5}$$

aggregated as $\text{Sharp}(\pi_{\text{rl}}) = \frac{1}{M} \sum_p \sigma(\pi_{\text{rl}}, p)$. $\hat{\pi}_{\text{rl}}(s \mid p)$ counts all rollouts using strategy s, not just correct ones; SHARPNESS measures strategy *selection*, not *execution*.

Strategy F1. Coverage is a corpus-level property ("can the model do geometry at all?"); sharpness is per-problem ("did it pick the right approach here?"). We combine them via their harmonic mean:

$$\text{STRATEGY-F1} = \frac{2 \cdot \text{Cov} \cdot \text{Sharp}}{\text{Cov} + \text{Sharp}}. \tag{6}$$

5 Experiments

We evaluate whether Strategy F1 distinguishes between RLVR methods in ways that pass@k cannot.

5.1 Setup

We evaluate five RLVR methods: R-Ada, KnapsackRL, GRESO, DAPO, and GRPO, all trained on top of Qwen2.5-Math-7b (Yang et al., 2024) with DAPO-Math-17k (Yu et al., 2026).

We evaluate the above models on AIME-2024 and AIME-2025. For each problem we generate multiple independent rollouts: $N = 256$ rollouts for the base model and $N = 64$ rollouts for each RL-trained model. Full details can be found in A.3

An LLM judge (gpt 5.4) classifies each solution into one of $K = 20$ strategy categories (full taxonomy in Appendix B). Once all the rollouts are assigned strategies, we apply our proposed metrics to compare models across the mentioned AIME datasets.

5.2 Results

Table 2 reports COVERAGE, SHARPNESS, Strategy F1, pass@1, and pass@64 for all five methods on both benchmarks.

F1 explains gaps that pass@k detects but cannot interpret. On AIME-2024 (Table 2, left), KnapsackRL achieves the highest pass@1 (29.9%) but the lowest pass@64 (56.7%); R-Ada achieves the lowest pass@1 (25.8%) but the highest pass@64 (70.0%). Pass@64 reveals this inversion but not its cause. Strategy F1 decomposes the gap: R-Ada retains more strategies (.947 vs. .895 coverage) and concentrates more on viable ones (.682 vs. .573 sharpness).

	AIME-2024					AIME-2025				
Model	Cov.	Sharp.	F1	pass@1	pass@64	Cov.	Sharp.	F1	pass@1	pass@64
R-Ada	**.947**	**.682**	**.793**	25.8	**70.0**	**.895**	**.531**	**.667**	**16.6**	**60.0**
KnapsackRL	.895	.573	.699	**29.9**	56.7	.895	.417	.569	14.9	40.0
GRESO	.789	.545	.645	28.7	60.0	.895	.449	.598	14.4	43.3
DAPO	.737	.618	.672	28.9	66.7	.895	.472	.618	16.4	46.7
GRPO	.737	.599	.661	29.2	63.3	.842	.423	.563	11.4	46.7

Table 2: Strategy F1 as the harmonic mean of reasoning diversity into coverage (retention of the base model's approach repertoire) and sharpness (concentration on viable approaches). R-Ada achieves the highest F1 on both benchmarks by balancing broad coverage with targeted sharpness. Pass rates shown as percentages.

The two axes diagnose different failures. In Table 2, DAPO and GRPO share the lowest coverage on AIME-2024 (.737) but DAPO's higher sharpness (.618 vs. .599) yields a 3.4-point pass@64 advantage. KnapsackRL retains broad coverage (.895) but low sharpness (.573); wide but unfocused. All three score within 1 point on pass@1. Coverage answers *what did RLVR lose?*; sharpness answers *how well does it select among what remains?*

Strategy F1 Identifies the Best Balance. If we look at only the coverage or only the sharpness, it might be difficult to gauge the balance of a RL method. Strategy F1 score helps us to determine the best balance between different methodologies. It effectively penalizes one-sided behavior. A model which is highly sharpened but has poor coverage receives a lower score similar to a model which has a high coverage but is poorly sharpened. In our experiments, R-Ada shows a good balance and hence it has a higher Strategy F1 score.

6 Analysis

The coverage–sharpness tradeoff in Table 2 is consistent with a positive feedback loop induced by RLVR. Verifiable rewards supervise final correctness, not the reasoning path used to reach the answer. Thus, distinct strategies can receive the same reward if they produce the correct final answer. Under on-policy sampling, strategies that are already sampled more frequently are also more likely to receive reward and contribute gradient signal, making them still more likely in later updates. Rare but viable strategies receive fewer opportunities for reinforcement and can gradually disappear.

This mechanism helps explain the sharper but lower-coverage behavior of methods such as GRPO. GRPO reinforces rollouts according to their relative advantage within a sampled group: responses that outperform the group mean are pushed up, while lower-reward alternatives are suppressed. When a small set of strategies already dominates the successful rollouts for a problem, this update can further concentrate probability mass on those strategies, improving sharpness while pruning parts of the base repertoire.

R-Ada appears to mitigate this effect by allocating more rollout budget to harder or lower-signal problems. This gives rare strategies more chances to appear, receive reward, and survive training, which is consistent with its higher coverage and Strategy F1. Overall, the desirable behavior is not maximum sharpening or maximum diversity alone, but selective preservation: retaining rare but viable strategies while increasing probability mass on strategies that solve each problem.

7 Conclusion

We introduced Strategy F1, a diagnostic that decomposes RLVR's effect on reasoning diversity into coverage and sharpness. Token entropy, the standard diversity metric, shows no consistent relationship with either single-sample accuracy or repeated-sampling gains; our decomposition does. Across five RLVR methods and two benchmarks, we find that all methods operate within the base model's existing strategy space, no new reasoning paths emerge, but navigate the coverage-sharpness tradeoff differently. Sharpness is the dominant predictor of test-time scaling: models that concentrate on their best viable strategy per problem benefit most from repeated sampling. Coverage independently diagnoses mode collapse, identifying which strategies RLVR discards. Together, the two axes provide a lens into RLVR behavior that pass@k alone cannot offer.

Limitations Our framework depends on an LLM judge (GPT-5.4) to classify reasoning strategies into $K{=}20$ categories. Misclassification introduces noise into both coverage and sharpness estimates; we report judge-human agreement in Appendix B but do not ablate sensitivity to taxonomy granularity; coarser or finer categories would change absolute metric values, though we expect relative rankings to remain stable. Coverage is set-based: a strategy observed once in $N{=}256$ samples receives the same credit as one observed 100 times, discarding frequency information that a distributional metric would capture. Viability is defined by a single correct solution ($\hat{c} > 0$), which is sensitive to sampling noise at small N. All five methods are trained on the same base model (Qwen2.5-Math-7b) and evaluated on competition mathematics (AIME); both the strategy taxonomy and the coverage–sharpness dynamics may differ for other base models, training corpora, or reasoning domains. Finally, our correlation analyses involve only five methods, limiting statistical power, we present the sharpness-scaling relationship as a consistent pattern and not as a confirmed statistical finding.

Ethical Considerations This work proposes an evaluation metric and does not introduce new models, training methods, or datasets. All models and benchmarks used are publicly available. Our strategy classification uses a commercial LLM (GPT-5.4), which may carry biases in how it labels reasoning approaches; we mitigate this by validating against human annotations. We see no direct negative societal impacts from this work.

References

Oron Anschel, Alon Shoshan, Adam Botach, Shunit Haviv Hakimi, Asaf Gendler, Emanuel Ben Baruch, Nadav Bhonker, Igor Kviatkovsky, Manoj Aggarwal, and Gerard Medioni. 2025. Group-aware reinforcement learning for output diversity in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32382–32403.

Ryan Bahlous-Boldi, Isha Puri, Idan Shenfeld, Akarsh Kumar, Mehul Damani, Sebastian Risi, Omar Khattab, Zhang-Wei Hong, and Pulkit Agrawal. 2026. Vector policy optimization: Training for diversity improves test-time search. *arXiv preprint arXiv:2605.22817*.

Sungmin Cha and Kyunghyun Cho. 2026. Why knowledge distillation works in generative models: A minimal working explanation. *Advances in Neural Information Processing Systems*, 38:30017–30037.

Xiwen Chen, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Hao Wang, Haiyu Wu, Huayu Li, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. 2025. Dra-grpo: Exploring diversity-aware reward adjustment for r1-zero-like training of large language models. *arXiv preprint arXiv:2505.09655*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, and 1 others. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Yuhua Jiang, Jiawei Huang, Yufeng Yuan, Xin Mao, Yu Yue, Qianchuan Zhao, and Lin Yan. 2025. Risk-sensitive rl for alleviating exploration dilemmas in large language models. *arXiv preprint arXiv:2509.24261*.

Changhao Li, Yuchen Zhuang, Chenxiao Gao, Haotian Sun, Rushi Qiang, Chao Zhang, and Bo Dai. 2026. Exploration-driven optimization for test-time large language model reasoning. *arXiv preprint arXiv:2605.09853*.

Ziniu Li, Congliang Chen, Tianyun Yang, Tian Ding, Ruoyu Sun, Ge Zhang, Wenhao Huang, and Zhi-Quan Luo. 2025. Knapsack rl: Unlocking exploration of llms via optimizing budget allocation. *arXiv preprint arXiv:2509.25849*.

Markan. n.d. AIME Syllabus. http://markan.net/aime/syllabus.pdf. Accessed: 2026-05-25.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. An empirical analysis of compute-optimal inference for problem-solving with language models.

Wei Xiong, Chenlu Ye, Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, Jiang Bian, Nan Jiang, and Tong Zhang. 2025. Reinforce-ada: An adaptive sampling framework under non-linear rl objectives. *arXiv preprint arXiv:2510.04996*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2017. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. *URL https://arxiv. org/abs/2504.13837*, 2.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, and 1 others. 2025. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. 2026. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Experimental Details

A.1 Datasets and Models

Tables 3, 4, and 5 summarize the datasets, benchmarks, and models used in this work.

Dataset	Problems	HuggingFace Url
DAPO-Math-17k	14,116	`BytedTsinghua-SIA/DAPO-Math-17k`

Table 3: Training datasets. DAPO-Math-17k contains 14,116 English problems after deduplication.

Benchmark	Problems	HuggingFace Url
AIME 2024	30	`Maxwell-Jia/AIME_2024`
AIME 2025	30	`MathArena/aime_2025`

Table 4: Evaluation benchmarks. SciKnowEval counts reflect the 10% test split.

Model	Params	HuggingFace Path
Qwen2.5-Math-7B (Yang et al., 2024)	7B	`Qwen/Qwen2.5-Math-7B`
Qwen2.5-Math-7B-GRPO	7B	Trained by us
Qwen2.5-Math-7B-DAPO	7B	Trained by us
Qwen2.5-Math-7B-GRESO	7B	Trained by us
Qwen2.5-Math-7B-KnapsackRL	7B	Trained by us
Qwen2.5-Math-7B-R-Ada (Xiong et al., 2025)	7B	`RLHFlow/Qwen2.5-Math-7B-Reinforce-Ada-balance-ha`

Table 5: Models used in our experiments. Qwen2.5-Math models are base (non-instruct) checkpoints.

A.2 Prompt

Prompt Template

```
<|im_start|>system
Please reason step by step, and put your final answer within \boxed{}.
<|im_end|>
<|im_start|>user
Please solve the following math problem: {{Question}}. The assistant first thinks about the reasoning process step by step and then provides the user with the answer. Return the final answer in \boxed{} tags, for example \boxed{1}. Let's solve this step by step.
<|im_end|>
<|im_start|>assistant
```

Figure 2: Prompt template used for training and evaluation. The system message is auto-injected by the Qwen2.5-Math tokenizer. The user message wraps the problem in the instruction format.

A.3 Training Setup

All the models were trained on $8\times$ H100 GPUs. We follow the available code released by the methods listed in our paper to train the models to be as close to the original methodology employed by the authors of the methods.

A.4 Sampling Parameters

For all the models, base and RL versions, we keep the temperature as 0.6 throughout with max tokens of 8192(extended with vLLM). We use vLLM for sampling the rollouts for each model on a node with H100 GPUs.

B AIME Taxonomy Construction

Our methodology requires assigning each rollout to a mathematical strategy it was solved with. To do this, we construct a strategy taxonomy for AIME-style problems. Starting from AIME syllabus (Markan, n.d.), we identify commom mathematical topics and solution approaches. We then use GPT-5.4 as an LLM Judge to validate, consolidate and refine the taxonomy into 20 distinct strategies.

We use this taxonomy to classify every rollout produced by each model. To do this, we prompt the judge, again GPT-5.4, with a rollout and the complete strategy set. The judge then assigns the rollout to the strategy that best characterizes its reasoning process. In cases where there is ambiguity, we use the judge's confidence scores to select the dominant strategies.

The buckets of strategies are described in table 6

Table 6: Strategy taxonomy for AIME problem-solving. Twenty strategies organized into four domains, each representing a distinct mathematical approach. The *Subsumes* column lists fine-grained techniques that fall under each strategy.

#	Strategy	Subsumes	Description
Algebra			
1	Algebraic Manipulation	substitutions, equation-solving, factorizations, logarithm rules, functional equations	Direct symbolic manipulation: setting up and solving equations, factoring, substitution, simplification
2	Polynomial Root Analysis	Vieta's formulas, Newton sums, polynomial transformations, Lagrange interpolation	Working with polynomials through their roots: relating roots to coefficients, power sums, interpolation
3	Sequences & Series	arithmetic/geometric/telescoping series, recursive sequences	Identifying and summing sequences, telescoping, finding closed forms
4	Inequalities & Optimization	AM-GM, Cauchy–Schwarz, rearrangement, bounding	Establishing bounds or extrema using classical inequalities
5	Complex Number Methods	complex arithmetic, roots of unity, polar form	Using complex number algebra to solve equations, sum trig values, or analyze periodicity
Geometry			
6	Synthetic Geometry	auxiliary lines, power of a point, similarity, triangle centers, Ptolemy's theorem, Menelaus–Ceva	Pure geometric reasoning: constructing auxiliary elements, applying circle/triangle theorems
7	Coordinate Geometry	coordinate bashing, Shoelace formula, distance formulas	Placing figures on a coordinate plane and computing with algebra
8	Trig/Complex Geometry	law of sines/cosines, trig Ceva, complex bashing, barycentric coordinates	Solving geometry via trigonometric or complex-number computation on geometric figures
9	3D Geometry	cross sections, surface unfolding, spatial visualization	Three-dimensional reasoning: cross sections, unfolding, shortest paths on surfaces
10	Geometric Transformations	rotation, reflection, inversion, moving points	Using symmetry, rotation, reflection, or inversion to simplify a geometric configuration
Combinatorics			
11	Direct Counting	combinations, permutations, stars and bars, constructive counting	Counting directly by choosing, arranging, or building valid configurations step-by-step
12	Casework & Enumeration	casework, binary representation, systematic enumeration	Splitting into exhaustive cases and counting each
13	Complementary Counting / PIE	complementary counting, inclusion–exclusion	Counting by subtracting the bad from the total, or using inclusion–exclusion
14	Recursion & Generating Fn.	recurrences, state machines, generating functions, Catalan/Fibonacci numbers	Setting up recurrences, state machines, or formal power series to count structured objects
15	Bijection & Symmetry	bijection, symmetry, invariants	Mapping to an easier-to-count set, exploiting symmetry, or identifying invariants
16	Probability & Expectation	linearity of expectation, geometric probability	Using probabilistic reasoning or linearity of expectation
Number Theory			
17	Modular Arithmetic	Euler's totient, Fermat's little theorem, CRT, quadratic residues	Computing in modular residues: order, Fermat/Euler, CRT
18	Prime Factorization / Div.	divisibility rules, LTE lemma, Zsigmondy's theorem	Analyzing divisibility structure through prime factorization or algebraic factorizations
19	Diophantine & Integer Eq.	Simon's Favorite Factoring Trick, Euclidean algorithm, Chicken McNugget theorem	Finding integer solutions: Bézout's identity, parametric families, bounding
20	Trigonometric Identities	sum-to-product, product-to-sum (non-geometric)	Pure trig identity manipulation: evaluating or simplifying trig expressions

C LLM Judge Prompts

Phase 1: Taxonomy construction. We use the syllabus from (Markan, n.d.), and prompt GPT-5.4 with the following prompt, to consolidate the syllabus into strategies that we can use for classification of rollouts.

Strategy Taxonomy Condensation Prompt

System: You are an expert in competition mathematics pedagogy. You will condense a fine-grained list of AIME problem-solving techniques into a compact taxonomy of high-level *strategies*.

Definitions:

nosep A **technique** is a specific mathematical tool (e.g., Vieta's formulas, AM-GM, Shoelace formula).

nosep A **strategy** is a broad problem-solving approach that may employ multiple techniques (e.g., "polynomial root analysis," "synthetic geometry").

Constraints:

nosep Organize strategies into four domains: Algebra, Geometry, Combinatorics, Number Theory.

nosep Each strategy must represent a *genuinely different* way to attack a problem — not a sub-step or minor variant.

nosep Every technique from the input list must map to exactly one strategy. List subsumed techniques explicitly.

nosep Two solutions should be assigned *different* strategies only if they rely on fundamentally different mathematical frameworks. Minor variations (different substitution, alternate factorization) belong to the same strategy.

nosep Provide a one-sentence description for each strategy explaining the core mathematical approach.

User: *[Full contents of the fine-grained technique list, organized by topic area]*

Condense these into approach-level strategies. For each strategy, list which techniques it subsumes and provide a one-sentence description.

Phase 2: Classification. Correct rollouts from each RL model are classified against the Phase 1 taxonomy:

Strategy Classification Prompt (System)

You are an expert mathematical reasoning classifier. You will classify math solutions into strategies from a fixed taxonomy.

TAXONOMY (N strategies):
[For each strategy in the taxonomy:]

- `strategy-name`: *description*

For each solution, assign exactly **ONE** strategy — the single most dominant mathematical method used. If a solution uses multiple techniques, pick the one that drives the overall approach.

Rate your **CONFIDENCE**: "high" (clear match), "medium" (reasonable but debatable), "low" (poor fit, forced choice).

Classify based on the core mathematical approach, **NOT** correctness, writing quality, or presentation.

Respond with JSON only:

```
{
  "classifications": [
    {"strategy": "strategy-name",
     "confidence": "high|medium|low"},
    ...
  ]
}
```

The classifications array must have exactly one entry per solution, in order. Strategy names must exactly match the taxonomy above.

Strategy Classification Prompt (User)

Problem: *[problem statement]*

=== Solution 1 ===
[full rollout text]

=== Solution 2 ===
[full rollout text]

...

Classify each solution's primary mathematical strategy from the taxonomy. Respond with JSON.

D RLVR Methods Occupy Different Coverage–Sharpness Regions.

We compare the methods through two parts of Strategy F1 - coverage and sharpness. Coverage measures how well does a model retain the coverage of the base's strategies after RL, while sharpness measures how strongly the model concentrates on viable strategies for each problem. Table 2 shows that the methods are not uniform across the measures, they occupy distinct coverage-sharpness regimes.

R-Ada stands out across methods as it has the highest coverage, highest sharpness and hence highest Strategy F1 across AIME-2024 and AIME-2025. It is able to preserve the highest coverage of strategies from the base model and also is substantially sharpened across them.

GRPO and DAPO show more aggressive sharpening on both the datasets. These methods shift the probability mass on more viable strategies, but this comes at a loss of coverage on the base strategy space. Hence, their strategy F1 is limited by lack of coverage.

KnapsackRL illustrates the opposite imbalance. It retains many base strategies, second to R-Ada, but the model is less concentrated on the viable ones. This suggests that preserving access alone is insufficient.

GRESO lies somewhere in between, it does not preserve as much coverage as R-Ada on AIME-2024 and does not sharpen as strongly as the best performing method, hence its F1 remains below R-Ada on both benchmarks.

E Per-Problem and Per-Strategy Analysis

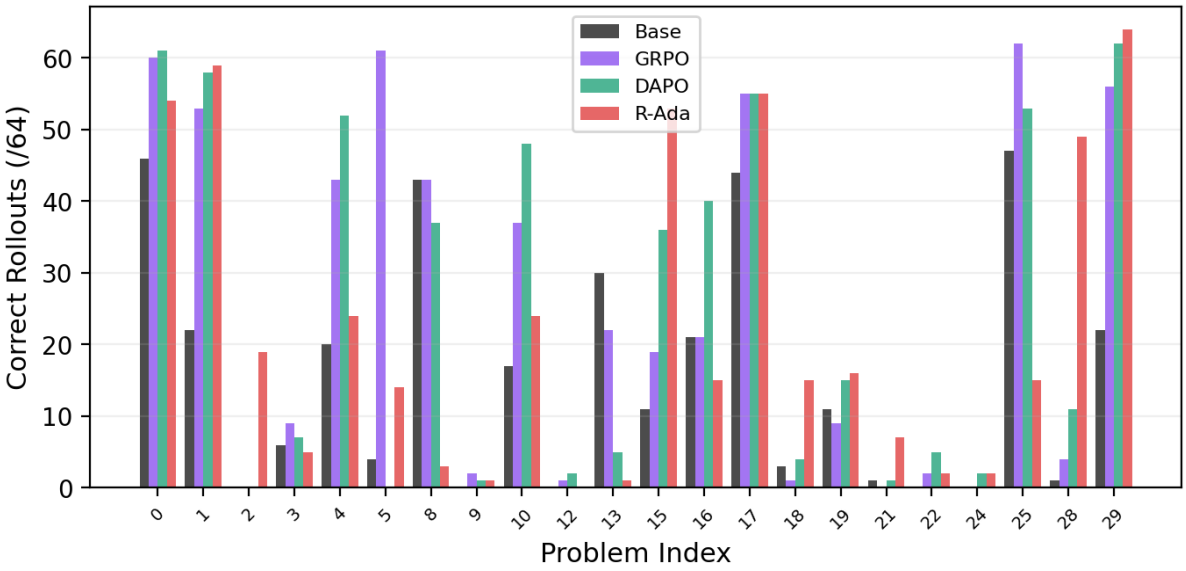

Figure 3: Correct rollouts per problem on AIME-2024 (N=64). R-Ada (red) matches or exceeds other methods on most problems and is the only method to solve Problem 2 (19/64 correct). GRPO and DAPO show sharper variation across problems, consistent with their lower coverage—they concentrate on fewer strategies and fail entirely on problems requiring approaches they have lost.

F Perplexity tracks strategy preference.

Computing each DAPO checkpoint's perplexity on rollouts from the two most dominant strategies, we find that by the final checkpoint the preferred strategy's perplexity drops below the alternative's (Figure 4). The narrowing reflects genuine shifts in the model's internal distribution, not merely a sampling artifact.

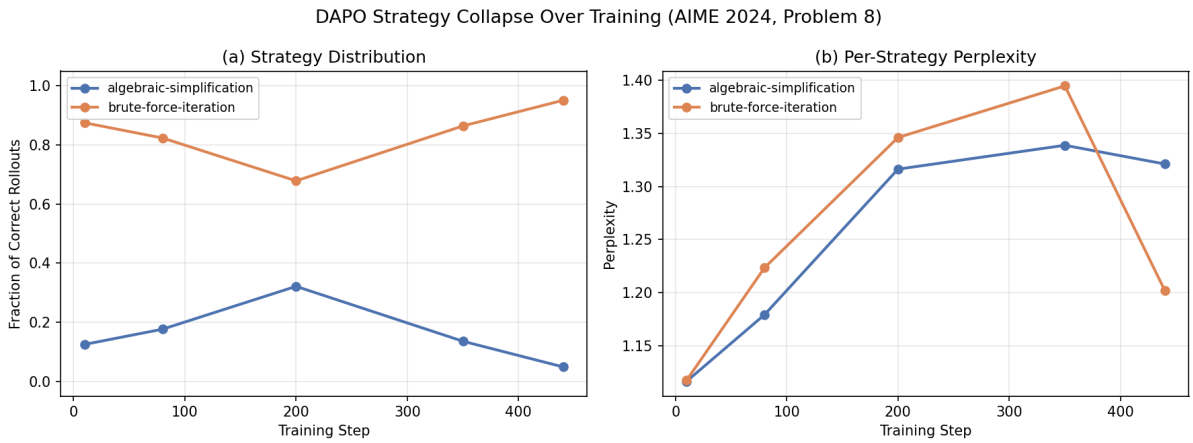

Figure 4: Strategy collapse during DAPO training on AIME-2024 Problem 8. **(a)** The fraction of correct rollouts using algebraic simplification rises from 0.35 to 0.85 over 400 training steps, while brute-force iteration drops from 0.65 to 0.15. **(b)** Per-strategy perplexity diverges: the model becomes increasingly fluent at the reinforced strategy and less fluent at the abandoned one. This illustrates the positive feedback loop from binary rewards and on-policy sampling: strategies that are sampled more frequently receive more reward signal, further increasing their probability at the expense of alternatives.

G Distribution of Strategies across Models

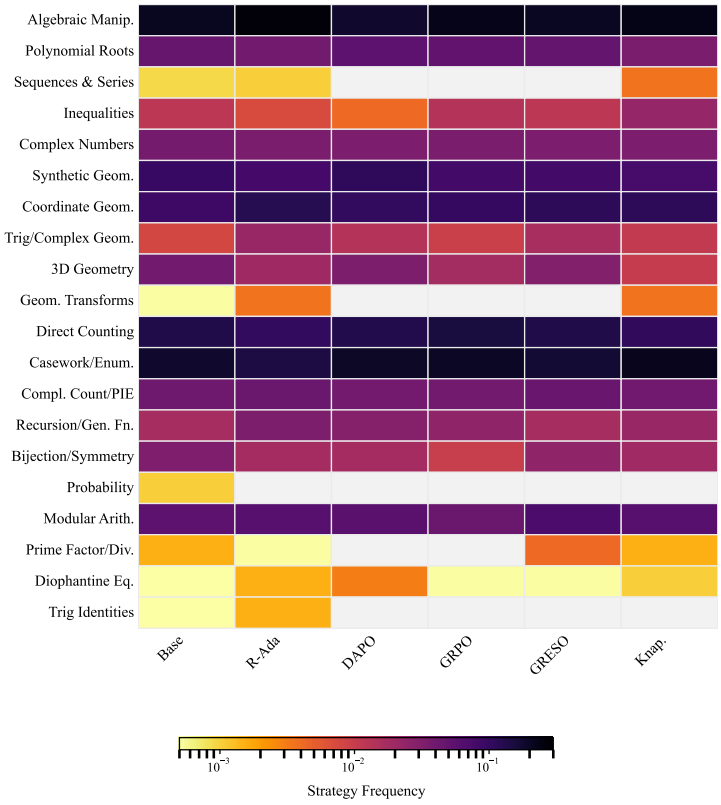

Figure 5: Here we show the frequency of usage of strategies across the models that we evaluated. We can see R-Ada shows the highest coverage and a good distribution of frequency across the strategy as we can interpret form the Strategy F1 score.