The Invisible Leash: Why RLVR May Not Escape Its Origin

Fang Wu [1 *] **Weihao Xuan** [2 3 *] **Ximing Lu** [4] **Zaid Harchaoui** [4] **Yejin Choi** [1 5]

Abstract

Recent advances in large reasoning models highlight Reinforcement Learning with Verifiable Rewards (RLVR) as a promising method for enhancing AI's capabilities, particularly in solving complex logical tasks. However, it remains unclear whether RLVR truly expands a model's reasoning boundary or merely amplifies high-reward outputs that the base model already knows for improved precision. This study presents a theoretical and empirical investigation that provides fresh insights into the potential limits of RLVR. First, we offer a new theoretical perspective that RLVR is constrained by the base model's support—unable to sample solutions with zero initial probability—and operates as a conservative reweighting mechanism that may restrict the discovery of entirely original solutions. We also identify an entropy–reward tradeoff: while RLVR reliably enhances precision, it may progressively narrow exploration and potentially overlook correct yet underrepresented solutions. Extensive empirical experiments validate that while RLVR consistently improves `pass@1`, *the shrinkage of empirical support generally outweighs the expansion of empirical support under larger sampling budgets*, failing to recover correct answers that were previously accessible to the base model. Interestingly, we also observe that while RLVR sometimes increases token-level entropy—resulting in greater uncertainty at each generation step—answer-level entropy declines, indicating that these seemingly more uncertain paths ultimately converge onto a smaller set of distinct answers. Taken together, these findings reveal potential limits of RLVR in extending reasoning horizons. Breaking this invisible leash may require future algorithmic innovations such as explicit exploration mechanisms or hybrid strategies that seed probability mass into underrepresented solution regions.

[*]Equal contribution [1]Department of Computer Science, Stanford University, USA [2]University of Tokyo, Japan [3]RIKEN AIP, Japan [4]University of Washington, USA [5]NVIDIA, USA. Correspondence to: Yejin Choi <yejinc@stanford.edu>.

The 2nd AI for MATH Workshop at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

1. Introduction

The rise of large reasoning models, such as DeepSeek-R1 (Guo et al., 2025) and OpenAI-o3 (Jaech et al., 2024), marks a breakthrough in AI capabilities, particularly in solving complex logical tasks involving mathematics (Luo et al., 2025c; Zeng et al., 2025) and programming (Luo et al., 2025b; Liu & Zhang, 2025). The key ingredient behind this remarkable progress is large-scale **Reinforcement Learning with Verifiable Rewards** (**RLVR**), where a pretrained base model—or one fine-tuned on long-form Chain-of-Thought (CoT) data—is optimized via reinforcement learning (RL) using simple, automatically computed rewards. Despite the empirical success, a fundamental question remains under active debate within the research community: *does RLVR expand a base model's reasoning capabilities, or does it simply reinforce patterns the base model already knows, sometimes at the expense of exploring alternative correct solutions?*

Recent studies offer divergent perspectives on this question. On the one hand, several works (Yue et al., 2025a; Zhao et al., 2025b; Shah et al., 2025; Ma et al., 2025; He et al., 2025) highlight a paradoxical failure mode: while RLVR-trained models outperform their base models on `pass@k` at low sampling budgets (e.g., $k = 1$), base models achieve higher `pass@k` scores when k is large, suggesting a narrowing of the reasoning horizon after RLVR training. Some even report that RLVR-trained models benefit from seemingly random or spurious reward signals (Shao et al., 2025), raising questions about whether the observed improvements genuinely reflect enhanced reasoning. On the other hand, Liu et al. (2025) report that previous studies focused primarily on special domains such as math, where base models may have been over-trained, which can then lead to premature termination of RLVR unless the level of entropy is carefully controlled. They then demonstrate that RLVR can expand the reasoning horizon considerably on certain domains, such as Reasoning Gym, where the base models struggle, with marked improvement on `pass@k` at large k.

While seeking a definitive answer to this debate remains an open challenge, we present a theoretical and empirical investigation that provides novel insights on the potential limits of RLVR, when using the currently competitive RLVR recipe. We first present a novel theoretical perspective that RLVR predominantly preserves the support of the base model. The intuition is that LLMs cannot sample solutions that have zero probability mass from the initial distribution, thus, the support of the base models inherently restricts the discovery of truly original reasoning patterns. Additionally, we provide a unified view of the RLVR objective via variational inference, revealing why RLVR is inherently conservative: it makes minimal updates to the base model's distribution, preserving relative probabilities within the reward-consistent subset. Lastly, we highlight the entropy-reward tradeoff: while RLVR reliably enhances precision, it may also progressively narrow the exploration of reasoning trajectories, potentially overlooking correct yet underrepresented solutions.

Empirically, we validate these theoretical insights via extensive experiments across diverse domains, including mathematics, logical reasoning, factual QA, and code generation tasks. To characterize RLVR's impact on the output distribution, we introduce the notion of *empirical support*—the set of correct completions assigned non-negligible probability under a model's sampling distribution. We find that while RLVR consistently improves `pass@1`, **the shrinkage of empirical support generally outweighs the expansion of empirical support under larger sampling budgets**. While RLVR can occasionally assign non-negligible probability mass to previously underrepresented correct completions (empirical-support expansion), we observe that the opposite—empirical-support shrinkage—is more frequent: RLVR often fails to recover correct answers that were previously accessible to the base model. This trend highlights RLVR's role as a conservative reweighting mechanism rather than a driver of fundamentally novel reasoning modes. To further understand how RLVR reshapes the sampling distribution, we decouple local uncertainty and global diversity via two entropy metrics: token-level and answer-level entropy. Interestingly, we observe that RLVR sometimes increases token-level entropy—reflecting greater uncertainty at each generation step—likely due to longer reasoning chains or more complex intermediate decisions. Yet, answer-level entropy declines, indicating that these more uncertain generation paths ultimately collapse onto a smaller set of distinct answers. This contrast reveals that RLVR models may appear more exploratory at the step level, even as they converge on fewer final completions.

Taken together, these findings suggest that there might exist inherent limits in RLVR for extending LLMs' reasoning horizons despite its empirical success. To break this invisible leash, RLVR may need augmenting with explicit exploration or hybrid strategies that seed probability mass into underrepresented regions of the solution space. We hope this work offers novel insights into RLVR's strengths and limitations, guiding future efforts in building LLM systems that can unlock genuinely new reasoning capacity.

2. Theoretical Limits of RLVR

2.1. Preliminaries

Let $\mathcal{X}$ denote the space of natural language prompts, and $\mathcal{Y}$ denote the space of token sequences (*e.g.*, reasoning traces or completions). For a fixed prompt $x \in \mathcal{X}$, $q(y \mid x)$ is the output distribution of the base model, and $R(x, y) \in \{0, 1\}$ is a verifiable reward function indicating whether y is a correct solution. Various RLVR algorithms, including PPO (Schulman et al., 2017), RLOO (Kool et al., 2019), GRPO (Guo et al., 2025), DAPO (Yu et al., 2025), or REINFORCE++ (Hu, 2025), learn a new distribution $\pi_\theta(y \mid x)$ to optimize different variants of the following regularized objective:

$$\max_\theta \mathbb{E}_{y\sim\pi_\theta(\cdot|x),x\sim\mathcal{D}}\left[R(x,y) - \beta^{-1}\log\frac{\pi_\theta(y \mid x)}{q(y \mid x)}\right], \tag{1}$$

where $\mathcal{D}$ is the distribution of prompts. An optional log ratio corresponds to a regularized policy update that penalizes divergence from the base model q controlled by a hyperparameter $\beta > 0$.

2.2. Support Preservation: Why RLVR Rarely Discovers New Modes

We begin by formalizing a core limitation of RLVR: it is inherently constrained to operate within the support of the base model's distribution. Since RLVR relies on gradient signals derived from samples generated by the base model, it cannot assign a nonzero probability to any solution that can never be sampled from $q(\cdot \mid x)$. As a result, any correct output y^* with $q(y^* \mid x) = 0$ remains inaccessible to policy gradient updates, regardless of reward.

Definition 2.1 (Support of Correct Completions)**.** *Let $\mathcal{C} = \{y \in \mathcal{Y} \mid R(x, y) = 1\}$ denote the set of correct completions under the reward function R. Then the* effective support on correct completions *of a distribution $p(y \mid x)$ is defined as*

$$\mathrm{supp}(p) := \{y \in \mathcal{C} \mid p(y \mid x) > 0\}.$$

We formalize this intuition with the following theorem, which makes precise how RLVR's reliance on the base model's sampling prevents discovering truly new solutions.

Theorem 2.2 (Support Preservation under RLVR)**.** *Let*

$\pi_\theta(y \mid x)$ be the RLVR-trained distribution obtained via standard on-policy gradient updates on verifiable rewards R. Then for all $x \in \mathcal{X}$,

$$\mathrm{supp}(\pi_\theta(\cdot \mid x)) \subseteq \mathrm{supp}(q(\cdot \mid x)).$$

In particular, if $q(y^* \mid x) = 0$ for some correct solution y^*, then RLVR cannot discover y^*.

Corollary 2.3 (Asymptotic Sampling Upper Bound)**.** *Let $\texttt{pass@}k_p(x)$ be the probability that at least one out of k samples $y_i \sim p(\cdot \mid x)$ is correct, i.e. $\texttt{pass@}k_p(x) = 1 - \big(\Pr_{y\sim p}[R(x,y) = 0]\big)^k$. Under the conditions of Thm. 2.2 and the sampling independence, we have*

$$\limsup_{k\to\infty} \texttt{pass@}k_{\pi_\theta}(x) \quad \le \quad \limsup_{k\to\infty} \texttt{pass@}k_q(x).$$

Those theorems formalize a critical limitation of RLVR: its optimization cannot expand the search space beyond the initial support of the base model. This limitation arises because on-policy sampling means the model updates only from what it already samples — lacking representational coverage means no gradient can ever push probability mass toward truly unseen solutions. Even when rewards provide a clear training signal, RLVR cannot access or discover solutions that the base model assigns zero probability. Proofs are in Appx. A.1 and A.2.

This manifests as a trade-off between sharpness and diversity: RLVR can improve `pass@1` by concentrating mass on known high-reward modes but tends to reduce `pass@k` performance for larger k, where broader coverage is beneficial. By contrast, the base model may occasionally sample correct answers from its long-tail distribution, giving it a statistical edge under high-k evaluations (Yue et al., 2025a; Liu et al., 2025). This asymptotic upper bound captures a ceiling: no matter how many samples are drawn, the RLVR-trained model cannot exceed the base model's `pass@k` in the limit.

Empirical-Support Relaxation. Thm. 2.2 assumes that q has exact zeros in its support and RLVR operates strictly on-policy. However, these conditions rarely hold in practice. Softmax layers yield strictly positive probabilities across all tokens, making the nominal support of q span the entire space $\mathcal{Y}$. This factor, along with sampling noise and or temperature scaling, contributes to what we refer to as *empirical support diffusion*: over time, the model may assign growing probability mass to completions that initially had negligible—but still nonzero—probability under the base model.

While $q(y \mid x)$ is technically positive for all y due to the softmax, many completions lie so deep in the tail that they are effectively invisible to the training algorithm under finite sampling. To formalize this, we develop relaxation and define the *empirical support under* ϵ as

$$\mathrm{supp}_\epsilon(q) := \{y \in \mathcal{C} \mid q(y \mid x) > \epsilon\},$$

where $\epsilon > 0$ denotes a small cutoff (e.g., 10^{-4}) that separates completions with practically observable likelihood from those that are statistically negligible. Completions outside this threshold are unlikely to be sampled in typical on-policy RL settings with finite rollouts. The choice of ϵ is thus crucial for assessing which completions are empirically reachable. Intuitively, ϵ should correspond to the minimum probability required for a correct completion to appear within k samples. We derive a principled estimate for this threshold based on sampling confidence bounds in Appx. A.7.

Definition 2.4 (Empirical-Support Expansion and Shrinkage)**.** *Given a threshold $\epsilon > 0$,*

- *We say RLVR achieves* empirical-support expansion under threshold ϵ *if* $\mathrm{supp}_\epsilon(\pi_\theta) \setminus \mathrm{supp}_\epsilon(q) \neq \emptyset$*, i.e. there exists at least one completion $y^* \in \mathcal{C}$ such that*

  $$q(y^* \mid x) \le \epsilon \quad \textit{but} \quad \pi_\theta(y^* \mid x) > \epsilon.$$

  That is, the RLVR-trained model assigns non-negligible probability mass to correct completions that were effectively negligible under the base model.

- *We say RLVR exhibits* empirical-support shrinkage under threshold ϵ *if* $\mathrm{supp}_\epsilon(q) \setminus \mathrm{supp}_\epsilon(\pi_\theta) \neq \emptyset$*, i.e. there exists at least one completion $y^* \in \mathcal{C}$ such that*

  $$q(y^* \mid x) > \epsilon \quad \textit{but} \quad \pi_\theta(y^* \mid x) \le \epsilon.$$

  This formalizes the phenomenon where RLVR concentrates probability mass onto a narrower subset of outputs, effectively excluding correct solutions that were previously accessible under the base model.

Recall $\mathcal{C}$ is the set of correct completions, and let $S := \mathcal{C} \setminus \mathrm{supp}_\epsilon(q)$ denote the set of low-density completions. We consider a single-step RLVR update of the form:

$$\pi_{\theta'}(\cdot \mid x) = (1-\gamma)\widetilde{\pi}_\theta(\cdot \mid x) + \gamma\, \pi_e(\cdot \mid x),$$

where $\widetilde{\pi}_\theta$ represents a reward-weighted distribution (e.g., exponential tilting), π_e denotes an exploration distribution, and $\gamma \in [0,1]$ is the mixing weight. This formula arises by considering an interpolation between a reward-tilted distribution $\tilde{\pi}_\theta$, which sharpens on known high-reward modes, and an

explicit exploration distribution π_e that seeds probability mass into underexplored regions. The mixing parameter γ controls the exploration-exploitation balance.

Theorem 2.5 (Empirical-Support Preservation)**.** *Given a fixed $\tau > 0$ and under the update rule above, if the preceding policy satisfies the KL budget $D_{\mathrm{KL}}(\pi_\theta \,\|\, q) \le \delta$, the probability mass that the updated policy assigns to any $y' \in S$ in the low–density tail obeys*

$$\pi_{\theta'}(y' \mid x) \le \gamma + (1-\gamma)\, e^{\beta} \left(\tau + \sqrt{2\delta}\right).$$

Thus, unless γ (exploration weight) or τ (tail threshold) is substantially large, the probability mass assigned to regions outside the base model's empirical support remains negligible.

In practice, RLVR algorithms typically impose strong KL regularization (small δ), and use conservative temperature settings (small β). These choices jointly control the amplification factor e^{β} and the additive tolerance $\sqrt{2\delta}$. When combined with minimal exploration (small γ), $\pi_{\theta'}(y' \mid x)$ remains negligible for completions in the low-density tail S. Consequently, RLVR tends to behave like *probability sharpening*—concentrating mass around the high-probability modes of q—rather than exploring or discovering entirely novel solutions. Overcoming this tendency requires explicit exploration mechanisms or off-policy data sources that intentionally seed mass into new regions. For instance, Xie et al. (2024) proposes an exploration-augmented preference optimization that addresses similar constraints in RL from human feedback (RLHF).

In this sense, RLVR inherits both the inductive biases and structural limitations of its initialization. Without deliberate intervention or scaling, it remains confined to the functional expressivity of the base model. Our framework formalizes why RLVR often improves sampling efficiency but rarely produces qualitatively new reasoning capabilities. We further explore these dynamics in the KL-free regime in Sec. 2.3, which clarifies how removing explicit regularization changes support behavior. Proof is provided in Appx. A.3.

2.3. A Variational and Conservative Policy Update

We now present a unified view of the RLVR objective through the lens of variational inference. This reveals why RLVR is inherently conservative: it makes minimal updates to the base distribution while ensuring improved performance.

Proposition 2.6 (KL Projection onto Reward-Consistent Distributions)**.** *Let $\Delta(\mathcal{Y})$ be the probability simplex over the finite output space $\mathcal{Y}$. Define the set of feasible policies that achieve at least a target expected reward ρ:*

$$\mathcal{P}_\rho := \{p(y \mid x) \in \Delta(\mathcal{Y}) \mid \mathbb{E}_p[R(x,y)] \ge \rho\}.$$

Then the solution to the variational problem, $\min_{\pi \in \mathcal{P}_\rho} \mathrm{KL}(\pi \,\|\, q)$, is the distribution within $\mathcal{P}_\rho$ that is closest in KL divergence to the base model. The optimal policy takes the form:

$$\pi^*(y \mid x) \propto q(y \mid x) \cdot \exp(\beta R(x,y)),$$

where $\beta \in \mathbb{R}_{\ge 0}$ is the dual variable associated with the reward constraint and $\beta = 0$ degenerates to the base policy q.

Notably, by standard convex duality, this solution also arises as the optimizer of the entropy-regularized problem $\max_{\pi \ll q} \mathbb{E}_\pi\big[R(x,y)\big] - \frac{1}{\beta}\mathrm{KL}\big(\pi \,\|\, q\big)$, which softens the constraint into a penalty. Thus, RLVR can be interpreted either as a *hard projection* onto the closest distribution achieving the reward target, or as a *soft trade-off* that balances expected reward with closeness to the base model. Similar exponential tilting policy improvement oracles have been analyzed in the context of KL-regularized contextual bandits and RLHF (Zhao et al., 2024), though their focus is on sample complexity under coverage.

KL-Free Limit. A relevant special case is the KL-free limit, where explicit KL regularization is removed ($\beta \to \infty$) (Wei et al., 2023; Yu et al., 2025; Luo et al., 2025a; Yue et al., 2025b). In this regime, RLVR simplifies to a hard-filtered projection onto reward-maximizing completions.

Corollary 2.7 (KL-Free Projection)**.** *In the limit $\beta \to \infty$, the RLVR update converges to the renormalized restriction of the base model to the correct completion set:*

$$\lim_{\beta \to \infty} \pi_\beta(y \mid x) \;=\; \frac{q(y \mid x)\, \mathbf{1}\{y \in \mathcal{C}\}}{\sum_{y' \in \mathcal{C}} q(y' \mid x)}.$$

Together, Prop. 2.6 and Cor. 2.7 illustrate a continuum of RLVR behaviors—from softly regularized reweighting (small β) to sharply constrained filtering (large β). Even in the KL-free limit, updates remain fundamentally anchored to the base model's distribution, preserving relative probabilities within the reward-consistent subset. Consequently, while this projection ensures stable, efficient updates, it inherently limits RLVR's exploratory capacity. As established in Thm. 2.2, RLVR remains confined to the initial support of the base model unless explicit mechanisms introduce meaningful probability mass to new regions. Thus, the variational interpretation clarifies RLVR's strengths in improving precision and efficiency within existing capabilities, alongside its limitations in fundamentally expanding model reasoning. A detailed proof is provided in Appx. A.4 and A.5.

2.4. Entropy–Reward Trade-off: Precision at the Cost of Answer Diversity

Another structural property of RLVR is its tendency to systematically reduce the entropy of the answer distribution.

Table 1. Empirical-support categorization across math reasoning benchmarks under high sampling budgets. Each completion is categorized by correctness and support status: **Preservation** indicates the solution is found by both base and ProRL; **Shrinkage** indicates the base model found it but ProRL did not; **Expansion** indicates only ProRL found it; and – denotes solutions found by neither. The bottom rows report the overall accuracy of each model on the corresponding benchmark.

Category	Correctness		AIME2024	AIME2025	AMC	Math	Minerva	Olympiad
	Base	ProRL	pass@8192	pass@8192	pass@8192	pass@8192	pass@8192	pass@8192
Preservation	✓	✓	23	20	39	494	173	600
Shrinkage	✓	✗	3	3	1	4	22	26
Expansion	✗	✓	0	0	0	0	0	3
–	✗	✗	4	7	0	2	77	46
Accuracy	Base		86.7%	76.7%	100%	99.6%	71.7%	92.7%
	ProRL		76.7%	66.7%	97.5%	98.8%	63.6%	89.3%

Table 2. Empirical-support categorization across non-math reasoning benchmarks.

Category	Correctness		SimpleQA	LiveBench-R	LiveBench-C	LiveBench-L	SciBench
	Base	ProRL	pass@512	pass@2048	pass@2048	pass@2048	pass@2048
Preservation	✓	✓	64	94	59	6	616
Shrinkage	✓	✗	20	6	17	5	35
Expansion	✗	✓	11	0	7	3	10
–	✗	✗	338	0	45	36	31
Accuracy	Base		19.4%	100.0%	59.4%	22.0%	94.1%
	ProRL		17.3%	94.0%	51.6%	18.0%	90.4%

This behavior arises naturally from reward optimization, which statistically favors sharper distributions concentrated on high-reward completions. While such entropy reduction is beneficial in domains like board games or math—where precision is paramount—it may also suppress valuable diversity in contexts that benefit from broader coverage or multiple valid outputs, such as story or dialogue generation (Chen et al., 2023) and coding copilots (Peng et al., 2023).

Theorem 2.8 (Entropy Reduction and Precision–Coverage Trade-off)**.** *Assume a finite output space $\mathcal{Y}$ and define the Shannon entropy of a distribution as $\mathcal{H}[p] := -\sum_{y\in\mathcal{Y}} p(y \mid x) \log p(y \mid x)$. Then the following statements hold:*

(a) ***Entropy reduction.*** *Any RLVR update π_θ satisfies*

$$\mathcal{H}[\pi_\theta] \;\leq\; \mathcal{H}[q],$$

with equality only if the reward is constant on the support of q.

(b) ***Trade-off with coverage.*** *Lower entropy increases sampling precision for small budgets, but for large k, reduces the diversity of explored outputs—potentially missing alternative correct completions.*

This trade-off underpins RLVR's empirical strengths in tasks with narrowly defined optimal solutions such as mathematical proofs or tactical game endgames (where precision is paramount), while also emphasizing the need for explicit diversity mechanisms in more open-ended domains such as code generation, creative writing (Feizi et al., 2023; Ding et al., 2024), or brainstorming (Chang & Li, 2025). Importantly, entropy reduction is not inherently undesirable: when a task admits a unique correct solution, lower answer-level entropy simply reflects desirable convergence. Importantly, even in multi-solution domains, concentrating mass on a narrower set may still be desirable under constrained compute budgets. However, our results show that entropy reduction can still lead to empirical-support shrinkage even in predominantly single-solution domains like math, where RLVR sometimes fails to recover valid completions still accessible to the more diverse base model. This highlights that entropy-induced narrowing is a general phenomenon, not limited to multi-solution tasks, underscoring the broader need for explicit exploration or diversity-promoting strategies. Complete proofs are provided in Appx. A.6.

3. Experiments

3.1. Evidence of Hidden-Support Dynamics

Setup. We adopt **ProRL** (Liu et al., 2025) as our RLVR method due to its robust long-horizon training framework. Starting from `DeepSeek-R1-Distill-Qwen-1.5B` as the base model, ProRL's `Nemotron-Research-Reasoning-Qwen-1.5B` leverages GRPO enhanced with decoupled clipping, dynamic sampling, KL divergence regularization, and periodic reference resets to sustain exploration and prevent

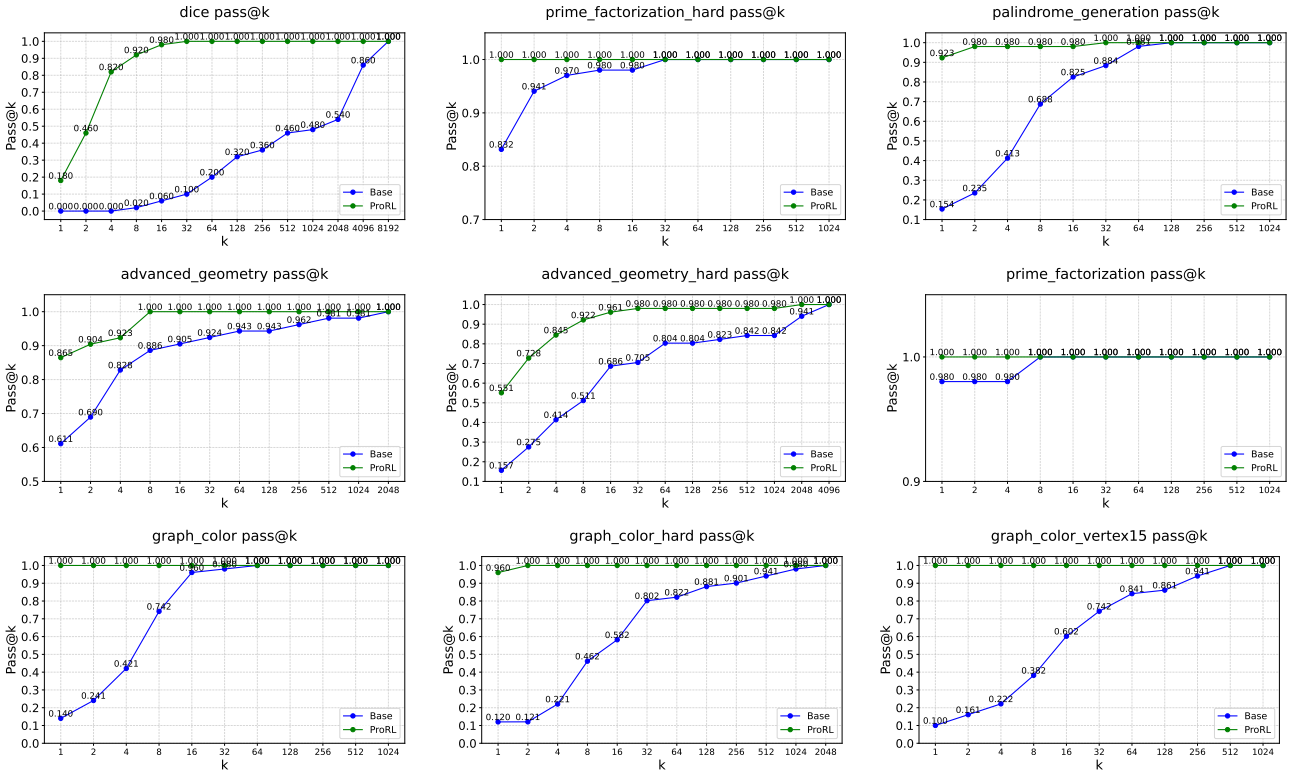

Figure 1. `Pass@k` curves on tasks like Graph Coloring, Palindrome Generation, and Advanced Geometry, illustrating RLVR's typical empirical-support preservation.

entropy collapse throughout prolonged RL training.

Performance is evaluated across two categories. (1) **math reasoning tasks** : MATH500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME 2024, AIME 2025, and AMC 2023. (2) **non-math reasoning tasks**: SimpleQA (Wei et al., 2024) (factuality), LiveBench (White et al., 2025) (logical reasoning, coding, language comprehension), SciBench (Wang et al., 2023) (multi-domain scientific problem-solving), and Reasoning Gym (Stojanovski et al., 2025) (cognition, geometry, graph theory, and common games). These benchmarks contain more general reasoning questions. In Reasoning Gym, we especially focus on tasks that ProRL explicitly highlighted as challenging for the base model. For SimpleQA, we use GPT-4.1 (Achiam et al., 2023) as the judge. The sampling is set at $k = 8192$ for math tasks, $k \in \{1024, 2048, 4096, 8192, 16384\}$ for Reasoning Gym, and $k \in \{512, 2048\}$ for non-math datasets, ensuring that any unreachable completion $y^* \in \mathcal{C}$ is below a pretty low threshold under empirical support of the base model. More detailed implementation is provided in Appendix B.

3.1.1. Results: Predominant Preservation with Limited Expansion

Support preservation dominates. Across most tasks, RLVR primarily sharpens the distribution within the effective support of the base model, aligning with our theoretical guarantees (Thm. 2.2 and 2.5). This is evident on Reasoning Gym tasks such as `graph_color` and `palindrome`, where RLVR accelerates convergence toward near-perfect `pass@k` under large sampling budgets (Fig. 1). Heatmaps and overlap counts in Tabs. 1 and 2 further highlight this predominant support preservation: for example, RLVR and the base model jointly recover 600 correct completions on Olympiad and 616 on SciBench, underscoring how RLVR chiefly reweights probability mass within the high-reward regions already represented by the base model.

Selective empirical-support expansion. Nonetheless, RLVR does occasionally assign non-negligible probability mass to completions that were effectively negligible under the base model's empirical support, uncovering genuinely new correct solutions. For instance, on OlympiadBench, it discovers 3 additional solutions; on SimpleQA and SciBench, 11 and 10, respectively. Likewise, Reasoning Gym tasks like `graph_color_vertex20` and `arc_1d`

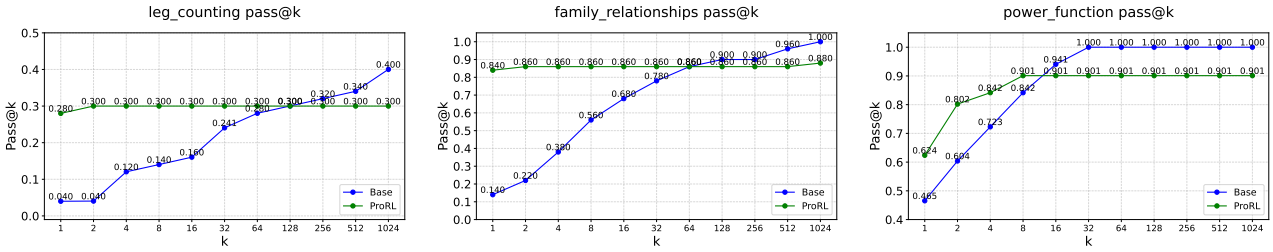

Figure 2. Examples of empirical-support shrinkage on Reasoning Gym tasks such as Leg Counting, Family Relationships, and Power Function.

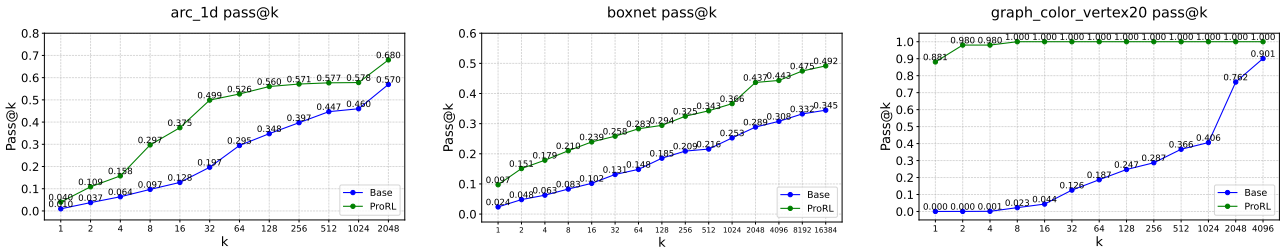

Figure 3. Rare instances of empirical-support expansion under RLVR, as seen in Boxnet, Dice, and Arc 1D tasks.

demonstrate striking empirical-support expansion (Fig. 3), where RLVR achieves near-perfect `pass@k` despite the base model struggling even under extensive sampling. These examples suggest that RLVR scaling may, at times, redistribute mass into underexplored solution modes, modestly broadening effective support.

Frequent empirical-support shrinkage. However, we find that empirical-support shrinkage—where RLVR fails to recover correct completions accessible to the base model—is even more pronounced. This aligns with the entropy-reducing, mode-concentrating effects predicted by Thm. 2.8. On the math benchmarks, RLVR misses 3 solutions on AIME2024, 3 on AIME2025, and experiences sharper losses on Minerva (22) and OlympiadBench (26). Similarly, it forfeits 20 and 35 correct completions on SimpleQA and SciBench, respectively. In Reasoning Gym, tasks such as `leg_counting`, `family_relationships`, and `power_function` illustrate this vividly: RLVR's distributions become markedly sharper (Fig. 2), rapidly saturating `pass@k` yet failing to explore alternative valid outputs that the more entropic base model uncovered.

Perplexity analysis on support constraints. Tab. 3 presents perplexity scores under two complementary settings. When evaluated against external reasoning traces from DeepSeek-R1 and Claude Sonnet 4 with extended thinking—likely outside the base model's support—RLVR shows markedly higher perplexity (e.g., AIME 2024 rising from 8.76 to 14.91), confirming that it cannot assign mass to fundamentally novel solution modes (Thm. 2.2). Note that differences in language style and reasoning format across external references (e.g., Claude vs DeepSeek) also contribute to perplexity gaps, beyond purely structural support constraints. Breakdowns by correctness patterns highlight the precision–coverage trade-off (Thm. 2.8): in shrinkage cases, ProRL's perplexity rises when it fails to recover solutions still accessible to the base, reflecting entropy-driven concentration. Meanwhile, the modest perplexity gaps in rare expansion cases indicate these *new* completions were actually drawn from the base's long-tail low-density regions—amplified but not truly beyond its support.

Overall takeaway: conservative optimization. A granular comparison reveals that *empirical-support shrinkage generally outweighs expansion.* Across Minerva and OlympiadBench, RLVR gains only 3 new completions but loses 48 previously found by the base model; on SimpleQA and SciBench, it gains 21 yet forfeits 55. Reasoning Gym presents a nuanced picture, with tasks like `boxnet` and `arc_1d` showing notable expansion, while others, such as `palindrome_generation_hard`, exhibit classic shrinkage patterns. Overall, these findings reinforce that RLVR chiefly acts as a sampling reweighting mechanism—concentrating probability mass within the existing representational landscape of the base model—offering higher precision but limited robust exploration. This resonates with the *Temporal Forgetting* phenomenon (Li et al., 2025), where fine-tuning often erases paths previously solvable at intermediate stages, affecting up to 56% of final

Table 3. Perplexity of reasoning tokens from base and ProRL models across math benchmarks, segmented by correctness patterns and reference types. Top: on problems correctly solved by the base model but not ProRL, perplexity is measured against the base model's reasoning traces. Middle: on problems correctly solved by ProRL but not the base, perplexity is measured against ProRL's traces. Bottom: on problems unsolved by both, perplexity is computed against external references (DeepSeek-R1 and Claude Sonnet 4), reflecting each model's compatibility with broader solution modes.

Correctness	Reference	Target	AIME 2024	AIME 2025	Olympiad
✓ Base, ✗ ProRL	Base	Base	1.36	1.47	1.30
		ProRL	1.60	1.84	1.50
✗ Base, ✓ ProRL	ProRL	Base	-	-	1.52
		ProRL	-	-	1.32
✗ Base, ✗ ProRL	DeepSeek-R1	Base	1.82	1.75	1.62
		ProRL	2.20	2.15	1.94
	Claude Sonnet 4	Base	8.76	6.05	5.98
		ProRL	14.91	9.76	9.55

failures. Together, our results underscore RLVR's role as a precision enhancer rather than a broad driver of novel reasoning discovery.

3.2. Entropy Reduction and the `pass@k` Trade-off

Setup. To study how RLVR reshapes the sampling distribution, we examine the base model and RLVR with a medium sampling budget $k = 32$ on the math reasoning benchmarks. We quantify changes in the output distribution using two entropy metrics:

- **Token-Level Entropy:** Let $\mathcal{V}$ denote the vocabulary and $y^{(i)} = (y_1^{(i)}, y_2^{(i)}, \ldots, y_{T^{(i)}}^{(i)})$ denote the i-th generated sequence of length $T^{(i)}$ for $1 \le i \le N$. At each timestep t, the model outputs a probability distribution $p_t^{(i)}(v)$ over vocabulary tokens $v \in \mathcal{V}$. The entropy of this distribution is given by: $H_t^{(i)} = -\sum_{v\in\mathcal{V}} p_t^{(i)}(v) \log p_t^{(i)}(v)$. The average token-level entropy over all N sequences and their timesteps is computed as: $\text{TokenEntropy} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T^{(i)}}\sum_{t=1}^{T^{(i)}} H_t^{(i)}\right)$, capturing the local uncertainty at each generation step.
- **Answer-level Entropy:** Let $\{o^{(1)}, \ldots, o^{(N)}\}$ denote the answers extracted from each generated sequence $y^{(i)}$ (using NA for incomplete outputs), and let $\{o_1^*, \ldots, o_M^*\}$ be the M unique answers. Let f_j be the frequency of answer o_j^*, with empirical probability $p_j = \frac{f_j}{N}$. Then: $\text{AnswerEntropy} = -\sum_{j=1}^{M} p_j \log p_j$. This captures global diversity over output completions, with lower values indicating increased mode collapse.

3.2.1. Results: Precision Gains, Entropy Dynamics, and Trade-offs

Consistent gains in precision, but sharper global distributions. As shown in Tab. 4, RLVR consistently improves `pass@1` accuracy across all benchmarks, raising average performance from 48.9% to 65.4%, which underscores its strength in reweighting probability mass toward high-reward completions and concentrating on likely correct answers (Dang et al., 2025). However, this increased precision comes at a cost: RLVR systematically reduces *answer-level entropy*, indicating a collapse onto fewer distinct solutions and empirically validating our theoretical prediction (Thm. 2.8) that reward optimization sharpens output distributions around known modes, thereby reducing effective support coverage. Notably, intrinsically harder tasks like AIME or Minerva still exhibit higher absolute answer-level entropy for both the base and RLVR models, suggesting that challenging problems inherently foster broader solution spaces requiring exploration over more diverse completions.

Decoupled local uncertainty and global diversity. Interestingly, while answer-level entropy consistently declines across all benchmarks, token-level entropy exhibits more varied behavior. In some models—such as ProRL and DAPO—it increases, suggesting greater local uncertainty during generation, possibly due to longer or more elaborated reasoning chains that introduce additional decision points or "forking" tokens (Wang et al., 2025). However, this pattern is far from universal: other RLVR models like AceReason display similar or even lower token-level entropy relative to their base counterparts, and prior work has documented sharp entropy collapse in early training phases (Cui et al., 2025). This disparity underscores that lower token-wise entropy is neither a necessary nor reliable outcome of RLVR training.

More importantly, increased token-level entropy does not imply greater exploration of the output space. Despite appearing more stochastic at the step level, RLVR models frequently converge onto a smaller set of final answers—reflected in lower answer-level entropy. This reveals a critical decoupling between local uncertainty and global

Table 4. Summary of `avg@32` accuracy, response length, and entropy metrics across math reasoning benchmarks (row colors: base models, RLVR models). RLVR consistently improves accuracy and alters distributional properties. While answer-level entropy consistently decreases, token-level entropy shows more varied behavior across models.

Metric	Model	AIME 2024	AMC 2023	MATH 500	Minerva	Olympiad	Avg.
avg@32 Acc. (%)	DeepSeek-1.5B	31.15	72.81	85.01	32.18	51.55	54.54
	ProRL-1.5B	45.62	85.70	92.01	39.27	64.56	65.43
	DeepSeek-7B	53.23	89.30	93.95	43.07	66.67	69.24
	AceReason-7B	65.83	95.08	95.81	45.35	73.92	75.20
	DeepSeek-14B	67.81	95.39	95.28	46.43	72.06	75.39
	AceReason-14B	77.29	98.67	97.01	47.20	77.74	79.58
	Qwen2.5-32B	18.12	55.23	75.84	24.55	41.40	43.03
	DAPO-32B	51.25	92.81	80.75	32.50	49.15	61.29
Response Length	DeepSeek-1.5B	16363	9979	5700	8194	11873	10422
	ProRL-1.5B	7786	6294	5070	6569	6678	6479
	DeepSeek-7B	13613	6402	4125	5595	8988	7745
	AceReason-7B	10740	5961	4313	6261	7703	6995
	DeepSeek-14B	11295	5735	3781	4919	8042	6755
	AceReason-14B	13871	7239	4622	7720	10033	8697
	Qwen2.5-32B	1247	874	585	3544	881	1426
	DAPO-32B	6908	3157	3386	5665	5827	4989
Token-Level Entropy	DeepSeek-1.5B	0.45	0.40	0.42	0.49	0.44	0.44
	ProRL-1.5B	0.47	0.51	0.54	0.55	0.52	0.52
	DeepSeek-7B	0.38	0.34	0.35	0.39	0.38	0.37
	AceReason-7B	0.18	0.23	0.27	0.24	0.23	0.23
	DeepSeek-14B	0.33	0.30	0.32	0.35	0.33	0.33
	AceReason-14B	0.12	0.13	0.15	0.15	0.14	0.14
	Qwen2.5-32B	0.17	0.16	0.15	0.28	0.15	0.18
	DAPO-32B	0.26	0.19	0.27	0.44	0.30	0.29
Answer-Level Entropy	DeepSeek-1.5B	2.15	0.91	0.46	1.65	1.33	1.30
	ProRL-1.5B	1.24	0.35	0.18	0.90	0.63	0.66
	DeepSeek-7B	1.47	0.36	0.18	0.96	0.80	0.75
	AceReason-7B	0.96	0.14	0.11	0.77	0.53	0.50
	DeepSeek-14B	1.01	0.14	0.13	0.83	0.59	0.54
	AceReason-14B	0.66	0.06	0.07	0.67	0.44	0.38
	Qwen2.5-32B	2.37	1.32	0.68	2.27	1.41	1.61
	DAPO-32B	1.12	0.09	0.26	0.96	0.63	0.61

diversity. We refer to this phenomenon as *local stochasticity without global exploration*: the model exhibits variability in generation but ultimately collapses to a narrow set of solutions. Thus, token-level entropy should not be conflated with genuine exploratory behavior, and interpreting entropy dynamics in RLVR requires distinguishing between stepwise uncertainty and overall support expansion.

Implications. Taken together, these findings reveal a fundamental trade-off in RLVR: it improves precision by amplifying high-reward outputs, but simultaneously narrows the diversity of global solutions. This limitation is especially consequential in domains that admit multiple valid answers or benefit from creative reasoning, underscoring the need for explicit exploration mechanisms or diversity-promoting strategies to complement standard RLVR. Moreover, the observed divergence between token-level and answer-level entropy highlights the need for a more nuanced interpretation of stochasticity in reward-optimized models—showing that precision gains often come at the expense of global diversity, and that maintaining controlled variability is critical for sustaining effective exploration.

4. Conclusion

We presented a unified theoretical and empirical analysis revealing that RLVR primarily acts as a conservative sampling reweighting mechanism: it improves precision by sharpening distributions around known high-reward trajectories, yet largely preserves the support of the base model. Importantly, we found that this sharpening does not merely prune incorrect outputs—it can also concentrate probability mass on a narrower subset of correct solutions, occasionally excluding valid alternatives that the more diverse base model could

still recover. This highlights a hidden trade-off between enhanced precision and comprehensive reasoning coverage. Notably, the divergence between token-level uncertainty and answer-level diversity also indicates that stepwise stochasticity alone is insufficient for global exploration, motivating future work to explicitly bridge this gap. Our findings suggest that to expand reasoning capabilities beyond the base model's scope truly, RLVR must be coupled with explicit exploration strategies or off-policy mechanisms that seed probability mass into underrepresented regions of the solution space.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Agarwal, S., Zhang, Z., Yuan, L., Han, J., and Peng, H. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*, 2025.

An, C., Xie, Z., Li, X., Li, L., Zhang, J., Gong, S., Zhong, M., Xu, J., Qiu, X., Wang, M., and Kong, L. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. URL https://hkunlp.github.io/blog/2025/Polaris.

Bae, S., Hong, J., Lee, M. Y., Kim, H., Nam, J., and Kwak, D. Online difficulty filtering for reasoning oriented reinforcement learning. *arXiv preprint arXiv:2504.03380*, 2025.

Chang, H.-F. and Li, T. A framework for collaborating a large language model tool in brainstorming for triggering creative thoughts. *Thinking Skills and Creativity*, pp. 101755, 2025.

Chen, G., Dong, S., Shu, Y., Zhang, G., Sesay, J., Karlsson, B. F., Fu, J., and Shi, Y. Autoagents: A framework for automatic agent generation. *arXiv preprint arXiv:2309.17288*, 2023.

Cui, G., Zhang, Y., Chen, J., Yuan, L., Wang, Z., Zuo, Y., Li, H., Fan, Y., Chen, H., Chen, W., et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Dang, X., Baek, C., Kolter, J. Z., and Raghunathan, A. Assessing diversity collapse in reasoning. In *Scaling Self-Improving Foundation Models without Human Supervision*, 2025.

Ding, S., Liu, Z., Dong, X., Zhang, P., Qian, R., He, C., Lin, D., and Wang, J. Songcomposer: A large language model for lyric and melody composition in song generation. *arXiv preprint arXiv:2402.17645*, 2024.

Feizi, S., Hajiaghayi, M., Rezaei, K., and Shin, S. Online advertisements with llms: Opportunities and challenges. *arXiv preprint arXiv:2311.07601*, 2023.

Guo, D., Yang, D., Zhang, H., and Song, J. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.07570*, 2025. URL https://arxiv.org/abs/2501.07570.

Habib, N., Fourrier, C., Kydlíček, H., Wolf, T., and Tunstall, L. Lighteval: A lightweight framework for llm evaluation, 2023. URL https://github.com/huggingface/lighteval.

He, A., Fried, D., and Welleck, S. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*, 2025.

He, C., Luo, R., Bai, Y., Hu, S., Thai, Z. L., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Hu, J. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

Jaech, A. et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Kool, W., van Hoof, H., and Welling, M. Buy 4 reinforce samples, get a baseline for free! *ICLR 2019 Workshop drlStructPred*, 2019.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Li, Y., Xu, Z., Jiang, F., Ramasubramanian, B., Niu, L., Lin, B. Y., Yue, X., and Poovendran, R. Temporal sampling for forgotten reasoning in llms. *arXiv preprint arXiv:2505.20196*, 2025.

Liu, J. and Zhang, L. Code-r1: Reproducing r1 for code with reliable rewards. 2025.

Liu, M., Diao, S., Lu, X., Hu, J., Dong, X., Choi, Y., Kautz, J., and Dong, Y. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.

Luo, M., Tan, S., Huang, R., Patel, A., Ariyak, A., Wu, Q., Shi, X., Xin, R., Cai, C., Weber, M., et al. Deepcoder: A fully open-source 14b coder at o3-mini level. *Notion Blog*, 2025a.

Luo, M., Tan, S., Huang, R., Shi, X., Xin, R., Cai, C., Patel, A., Ariyak, A., Wu, Q., Zhang, C., Li, L. E., Popa, R. A., and Stoica, I. Deepcoder: A fully open-source 14b coder at o3-mini level. https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51, 2025b. Notion Blog.

Luo, M., Tan, S., Wong, J., Shi, X., Tang, W. Y., Roongta, M., Cai, C., Luo, J., Li, L. E., Popa, R. A., and Stoica, I. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025c. Notion Blog.

Ma, L., Liang, H., Qiang, M., Tang, L., Ma, X., Wong, Z. H., Niu, J., Shen, C., He, R., Cui, B., et al. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. *arXiv preprint arXiv:2506.07527*, 2025.

Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590*, 2023.

Prabhudesai, M., Chen, L., Ippoliti, A., Fragkiadaki, K., Liu, H., and Pathak, D. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shafayat, S., Tajwar, F., Salakhutdinov, R., Schneider, J., and Zanette, A. Can large reasoning models self-train? *arXiv preprint arXiv:2505.21444*, 2025.

Shah, D. J. et al. Rethinking reflection in pre-training. *arXiv preprint arXiv:2504.04022*, 2025.

Shao, R., Li, S. S., Xin, R., Geng, S., Wang, Y., Oh, S., Du, S. S., Lambert, N., Min, S., Krishna, R., Tsvetkov, Y., Hajishirzi, H., Koh, P. W., and Zettlemoyer, L. Spurious rewards: Rethinking training signals in rlvr, 2025. Notion Blog.

Stojanovski, Z., Stanley, O., Sharratt, J., Jones, R., Adefioye, A., Kaddour, J., and Köpf, A. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards, 2025. URL https://arxiv.org/abs/2505.24760.

Wang, S., Yu, L., Gao, C., Zheng, C., Liu, S., Lu, R., Dang, K., Chen, X., Yang, J., Zhang, Z., et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

Wang, X., Hu, Z., Lu, P., Zhu, Y., Zhang, J., Subramaniam, S., Loomba, A. R., Zhang, S., Sun, Y., and Wang, W. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023.

Wei, J., Karina, N., Chung, H. W., Jiao, Y. J., Papay, S., Glaese, A., Schulman, J., and Fedus, W. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

Wei, T., Zhao, L., Zhang, L., Zhu, B., Wang, L., Yang, H., Li, B., Cheng, C., Lü, W., Hu, R., et al. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*, 2023.

White, C., Dooley, S., Roberts, M., et al. Livebench: A challenging, contamination-limited llm benchmark. In *International Conference on Learning Representations (ICLR)*, 2025.

Xie, T., Foster, D. J., Krishnamurthy, A., Rosset, C., Awadallah, A., and Rakhlin, A. Exploratory preference optimization: Harnessing implicit q*-approximation for sample-efficient rlhf. *arXiv preprint arXiv:2405.21046*, 2024.

Xiong, W., Yao, J., Xu, Y., Pang, B., Wang, L., Sahoo, D., Li, J., Jiang, N., Zhang, T., Xiong, C., et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

Xu, Y. E., Savani, Y., Fang, F., and Kolter, Z. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*, 2025.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Fan, T., Liu, G., Liu, L., Liu, X., et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025a. URL https://arxiv.org/abs/2504.13837.

Yue, Y., Yuan, Y., Yu, Q., Zuo, X., Zhu, R., Xu, W., Chen, J., Wang, C., Fan, T., Du, Z., et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025b.

Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., and He, J. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

Zhang, Q., Wu, H., Zhang, C., Zhao, P., and Bian, Y. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025a.

Zhang, X. et al. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*, 2025b.

Zhao, A. et al. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335*, 2025a.

Zhao, H., Ye, C., Gu, Q., and Zhang, T. Sharp analysis for kl-regularized contextual bandits and rlhf. *arXiv preprint arXiv:2411.04625*, 2024.

Zhao, R. et al. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. *arXiv preprint arXiv:2504.07912*, 2025b.

Zhao, X., Kang, Z., Feng, A., Levine, S., and Song, D. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025c.

Zhu, X., Xia, M., Wei, Z., Chen, W.-L., Chen, D., and Meng, Y. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025.

Zuo, Y., Zhang, K., Qu, S., Sheng, L., Zhu, X., Qi, B., Sun, Y., Cui, G., Ding, N., and Zhou, B. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

A. Mathematical Analysis

A.1. Proof of Thm. 2.2

Base case. By construction we initialize the RLVR policy to the base model:

$$\pi_{\theta_0}(y \mid x) \;=\; q(y \mid x).$$

Hence

$$\operatorname{supp}\big(\pi_{\theta_0}(\cdot \mid x)\big) = \operatorname{supp}\big(q(\cdot \mid x)\big).$$

Inductive step. Assume that at some iteration θ we have

$$\pi_\theta(y^* \mid x) \;=\; 0 \quad \text{for a particular } y^*.$$

All standard policy-gradient updates (e.g. REINFORCE, PPO, GRPO) take the form

$$\theta' \;=\; \theta + \eta\, \nabla_\theta \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[R(x,y) \;-\; \beta^{-1} \log \tfrac{\pi_\theta(y|x)}{q(y|x)}\Big].$$

Since the outer expectation is over $y \sim \pi_\theta$, any $y^* \in \mathcal{C}$ with $\pi_\theta(y^* \mid x) = 0$ is never sampled and thus contributes no gradient component. Therefore

$$\pi_{\theta'}(y^* \mid x) \;=\; 0,$$

and the support of $\pi_{\theta'}$ remains a subset of that of q.

Conclusion. By induction, none of the updates can introduce positive probability mass on any $y^* \in \mathcal{C}$ for which $q(y^* \mid x) = 0$. Equivalently,

$$\operatorname{supp}\big(\pi_\theta(\cdot \mid x)\big) \;\subseteq\; \operatorname{supp}\big(q(\cdot \mid x)\big),$$

indicating that any correct solution y^* with $q(y^* \mid x) = 0$ remains unreachable by the RLVR policy.

■

A.2. Proof of Corollary 2.3

From Thm. 2.2, support preservation implies

$$\operatorname{supp}(\pi_\theta(\cdot|x)) \;\subseteq\; \operatorname{supp}(q(\cdot|x)).$$

Thus, for any $y \in C$,

$$\pi_\theta(y|x) > 0 \;\implies\; q(y|x) > 0.$$

Define the total mass on correct completions by

$$\pi_\theta(C) \;=\; \Pr_{y\sim\pi_\theta}[R(x,y)=1], \quad q(C) \;=\; \Pr_{y\sim q}[R(x,y)=1].$$

Here, we assume samples are independent across LLMs' different draws; otherwise, we can only assert an upper bound by union bounds. As $k \to \infty$, the `pass@k` success probability to be written as

$$\texttt{pass@}k_{\pi_\theta}(x) = 1 - (1 - \pi_\theta(C))^k \;\longrightarrow\; \begin{cases} 1, & \pi_\theta(C) > 0, \\ 0, & \pi_\theta(C) = 0, \end{cases}$$

and similarly for q.

Because support preservation ensures that any correct completion reachable under π_θ must also be reachable under q,

$$\pi_\theta(C) > 0 \;\implies\; q(C) > 0.$$

Hence, the asymptotic success probability satisfies

$$\lim_{k\to\infty} \texttt{pass@}k_{\pi_\theta}(x) \;\le\; \lim_{k\to\infty} \texttt{pass@}k_q(x).$$

■

A.3. Proof of Thm. 2.5

We consider the one-step RLVR policy update given by:

$$\pi_{\theta'}(\cdot \mid x) = (1-\gamma)\,\widetilde{\pi}_\theta(\cdot \mid x) + \gamma\,\pi_e(\cdot \mid x),$$

where $\widetilde{\pi}_\theta(y \mid x) \propto \pi_\theta(y \mid x) \cdot \exp(\beta\, R(x,y))$ is the exponentially tilted distribution. $\pi_e(\cdot \mid x)$ is an arbitrary exploration distribution, $\gamma \in [0,1]$ is the mixing coefficient, and $\beta > 0$ is the inverse temperature.

We aim to bound $\pi_{\theta'}(y' \mid x)$ for any $y' \in S$. By the RLVR update rule and since $\pi_e(y' \mid x) \le 1$,

$$\pi_{\theta'}(y' \mid x) = (1-\gamma)\,\widetilde{\pi}_\theta(y' \mid x) + \gamma\,\pi_e(y' \mid x) \le (1-\gamma)\,\widetilde{\pi}_\theta(y' \mid x) + \gamma.$$

Bound on $\widetilde{\pi}_\theta(y)$. Let $Z = \sum_{y \in \mathcal{Y}} \pi_\theta(y \mid x)\, \exp(\beta\, R(x,y))$ be the normalizing constant of the exponential tilting. Since $\exp(\beta\, r) \ge 1$, $Z \ge \sum_{y \in \mathcal{Y}} \pi_\theta(y \mid x) = 1$. Then for all $y' \in S$ and $R(x,y) \in [0,1]$, it implies

$$\widetilde{\pi}_\theta(y' \mid x) = \frac{\pi_\theta(y' \mid x) e^{\beta R(x,y')}}{Z} \le \pi_\theta(y' \mid x) e^{\beta R(x,y')} \le \pi_\theta(y' \mid x) e^{\beta}.$$

Tail mass of the current policy. Now we bound $\pi_\theta(y' \mid x)$ in terms of $q(y' \mid x)$. Using Pinsker's inequality:

$$\mathrm{KL}(\pi_\theta \| q) \le \delta \quad \Rightarrow \quad \|\pi_\theta - q\|_1 \le \sqrt{2\delta} \quad \Rightarrow \quad \big|\pi_\theta(y' \mid x) - q(y' \mid x)\big| \le \sqrt{2\delta}.$$

Thus, for all $y' \in S$,

$$\pi_\theta(y' \mid x) \le q(y' \mid x) + \sqrt{2\delta} \le \tau + \sqrt{2\delta}.$$

Final bound. Combining the above gives

$$\widetilde{\pi}_\theta(y' \mid x) \le e^{\beta} \cdot (\tau + \sqrt{2\delta}),$$

and so

$$\boxed{\pi_{\theta'}(y' \mid x) \le (1-\gamma)\, e^{\beta}(\tau + \sqrt{2\delta}) + \gamma}.$$

■

A.4. Proof of Prop. 2.6

We provide two closely related derivations to illuminate the same optimal solution from both a hard-constrained and a soft-regularized perspective.

Convexity of Feasible Set P_ρ. We first prove the convexity of P_ρ. Recall $P_\rho = \Big\{p \in \Delta(\mathcal{Y}) : \sum_y p(y) R(x,y) \ge \rho\Big\}$, where $\Delta(\mathcal{Y})$ denotes the probability simplex over $\mathcal{Y}$.

Take any two distributions $p_1, p_2 \in P_\rho$ and let $\lambda \in [0,1]$. Consider the convex combination

$$p_\lambda := \lambda p_1 + (1-\lambda) p_2.$$

Since $\Delta(\mathcal{Y})$ is convex, we have $p_\lambda \in \Delta(\mathcal{Y})$.

Next, because $p_1, p_2 \in P_\rho$, it follows that

$$\sum_y p_1(y) R(x,y) \ge \rho \quad \text{and} \quad \sum_y p_2(y) R(x,y) \ge \rho.$$

Thus,

$$\sum_y p_\lambda(y) R(x,y) = \lambda \sum_y p_1(y) R(x,y) + (1-\lambda) \sum_y p_2(y) R(x,y) \ge \lambda\rho + (1-\lambda)\rho = \rho.$$

Hence $p_\lambda \in P_\rho$. This shows that P_ρ is convex.

Convexity, existence, and strong duality. We then verify the foundational properties of the optimization problem. Recall we wish to solve

$$\min_{\pi \in P_\rho} \mathrm{KL}(\pi \| q), \quad \text{where } P_\rho = \left\{ \pi \in \Delta(\mathcal{Y}) : \sum_y \pi(y) R(x,y) \geq \rho \right\}.$$

The objective function $\mathrm{KL}(\pi \| q)$ is convex in π over the probability simplex $\Delta(\mathcal{Y})$, since relative entropy is jointly convex and thus convex in π for fixed q. The feasible set P_ρ is also convex.

Moreover, if there exists a strictly feasible distribution π such that $\sum_y \pi(y) R(x,y) > \rho$, then by *Slater's condition*, strong duality holds. This guarantees that the optimal value of the primal problem equals the optimal value of its Lagrangian dual, and the Karush-Kuhn-Tucker (KKT) conditions characterize the optimal solution. In typical applications—where q arises from softmax-based models with full support—such strictly feasible distributions exist, ensuring that our subsequent Lagrangian approach is valid.

1) Hard-constrained formulation (projection perspective). Consider the optimization problem:

$$\min_\pi \mathrm{KL}(\pi \| q) \quad \text{s.t.} \quad \mathbb{E}_\pi[R(x,y)] \geq \rho, \quad \sum_y \pi(y \mid x) = 1, \quad \pi(y \mid x) \geq 0.$$

Using the method of Lagrange multipliers, the Lagrangian is:

$$\mathcal{L}(\pi, \beta, \lambda) = \sum_y \pi(y \mid x) \log \frac{\pi(y \mid x)}{q(y \mid x)} - \beta \left(\sum_y \pi(y \mid x) R(x,y) - \rho \right) + \lambda \left(\sum_y \pi(y \mid x) - 1 \right).$$

Here, we compute the derivative concerning $\pi(y \mid x)$ for fixed multipliers, thereby finding the stationary points of the Lagrangian. Specifically, we take derivative with respect to $\pi(y \mid x)$ and set it to zero:

$$\log \frac{\pi(y \mid x)}{q(y \mid x)} + 1 - \beta R(x,y) + \lambda = 0.$$

Solving for π yields:

$$\pi(y \mid x) \propto q(y \mid x) \cdot \exp(\beta R(x,y)).$$

2) Soft-regularized formulation (dual perspective). Alternatively, assume RLVR solves the entropy-regularized objective

$$\pi_\theta = \arg\max_{\pi \ll q} \mathbb{E}_{y \sim \pi}[R(x,y)] - \beta^{-1} \mathrm{KL}(\pi \,\|\, q),$$

for some inverse temperature parameter $\beta > 0$. Here, the constraint $\pi \ll q$ denotes that π is absolutely continuous with respect to q, meaning $\pi(y \mid x) > 0$ only if $q(y \mid x) > 0$.[1] The objective is equivalent to the following minimization:

$$\pi_\theta = \arg\min_{\pi \in \Delta(\mathcal{Y})} \mathrm{KL}(\pi \,\|\, q) - \beta\, \mathbb{E}_{y \sim \pi}[R(x,y)].$$

The Lagrangian becomes

$$\mathcal{L}(\pi, \lambda) = \sum_{y \in \mathcal{Y}} \pi(y) \log \frac{\pi(y)}{q(y)} - \beta \sum_{y \in \mathcal{Y}} \pi(y) R(x,y) + \lambda \left(\sum_{y \in \mathcal{Y}} \pi(y) - 1 \right),$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier enforcing the normalization constraint.

Taking the derivative with respect to $\pi(y)$ and setting it to zero:

$$\frac{\partial \mathcal{L}}{\partial \pi(y)} = \log \frac{\pi(y)}{q(y)} + 1 - \beta R(x,y) + \lambda = 0.$$

[1] Formally, absolute continuity $\pi \ll q$ ensures that the KL divergence $\mathrm{KL}(\pi \,\|\, q)$ is finite. If π assigns positive mass to any output that q assigns zero probability, the divergence becomes infinite. This condition also enforces support preservation: $\mathrm{supp}(\pi) \subseteq \mathrm{supp}(q)$.

Solving for $\pi(y)$ gives:

$$\pi(y) = q(y) \cdot \exp\left(\beta R(x, y) - \lambda - 1\right).$$

Letting the normalization constant be:

$$Z = \sum_{y' \in \mathcal{Y}} q(y') \cdot \exp(\beta R(x, y')),$$

we absorb constants into Z and write:

$$\pi_\theta(y \mid x) = \frac{q(y \mid x) \cdot \exp(\beta R(x, y))}{Z}.$$

Both derivations recover the same *exponentially tilted* distribution that emphasizes high-reward completions relative to the base model. In the hard-constrained view, β is a Lagrange multiplier tuned to meet the target reward ρ; in the soft-regularized view, β sets the strength of the trade-off between reward and divergence. This completes the constructive proof of Prop. 2.6.

■

A.5. Proof of Cor. 2.7

Since $R(x, y) \in \{0, 1\}$, we have

$$\exp\big(\beta R(x, y)\big) = \begin{cases} e^\beta & \text{if } R(x, y) = 1, \\ 1 & \text{if } R(x, y) = 0. \end{cases}$$

Thus the RLVR distribution becomes

$$\pi_\beta(y \mid x) = \frac{q(y \mid x)\, \exp\big(\beta R(x, y)\big)}{Z_\beta(x)} = \frac{q(y \mid x)\, \left[e^\beta \mathbf{1}\{R(x, y) = 1\} + \mathbf{1}\{R(x, y) = 0\}\right]}{Z_\beta(x)},$$

where

$$Z_\beta(x) = e^\beta \sum_{y' : R(x, y') = 1} q(y' \mid x) + \sum_{y' : R(x, y') = 0} q(y' \mid x).$$

As $\beta \to \infty$, the term with e^β dominates whenever there exists at least one y with $R(x, y) = 1$. Thus

$$Z_\beta(x) \approx e^\beta \sum_{y' \in \mathcal{C}} q(y' \mid x).$$

Similarly, in the numerator we have

$$q(y \mid x)\, \exp\big(\beta R(x, y)\big) = \begin{cases} q(y \mid x)\, e^\beta & \text{if } y \in \mathcal{C}, \\ q(y \mid x) & \text{otherwise.} \end{cases}$$

Dividing by $Z_\beta(x)$ and taking $\beta \to \infty$, the probabilities assigned to y with $R(x, y) = 0$ vanish:

$$\pi_\beta(y \mid x) \approx \begin{cases} \dfrac{q(y \mid x)\, e^\beta}{e^\beta \sum_{y' \in \mathcal{C}} q(y' \mid x)} = \dfrac{q(y \mid x)}{\sum_{y' \in \mathcal{C}} q(y' \mid x)} & \text{if } y \in \mathcal{C}, \\ 0 & \text{otherwise.} \end{cases}$$

Thus we obtain

$$\lim_{\beta \to \infty} \pi_\beta(y \mid x) = \frac{q(y \mid x)\, \mathbf{1}\{y \in \mathcal{C}\}}{\sum_{y' \in \mathcal{C}} q(y' \mid x)},$$

A.6. Proof of Thm. 2.8

(a) Entropy reduction. Consider the exponentially tilted distribution

$$\pi_\theta(y \mid x) = \frac{q(y \mid x)\exp(\beta R(x,y))}{Z}, \quad \text{with} \quad Z = \sum_{y \in \mathcal{Y}} q(y \mid x)\exp(\beta R(x,y)).$$

By standard properties of KL divergence,

$$\mathrm{KL}(\pi_\theta \| q) = \sum_y \pi_\theta(y \mid x) \log \frac{\pi_\theta(y \mid x)}{q(y \mid x)} \quad \geq \quad 0.$$

Rearranging gives

$$\mathcal{H}[\pi_\theta] = \mathcal{H}[q] - \mathrm{KL}(\pi_\theta \| q) \quad \leq \quad \mathcal{H}[q].$$

Thus, any such RLVR update decreases entropy relative to the base distribution, unless the reward is constant (in which case $\pi_\theta = q$).

(b) Trade-off with diversity at different sampling budgets. The RLVR-trained policy sharpens the probability mass around high-reward completions. Explicitly,

$$\pi_\theta(y \mid x) \;\propto\; q(y \mid x)\exp(\beta R(x,y)),$$

where $\beta > 0$ controls concentration.

- **Small sampling budgets ($k = 1$):** The increased probability on high-reward outputs generally improves single-shot success rates. Formally,

  $$\texttt{pass@1}_{\pi_\theta}(x) = \sum_{y:R(x,y)=1} \pi_\theta(y \mid x) \quad > \quad \sum_{y:R(x,y)=1} q(y \mid x) = \texttt{pass@1}_q(x),$$

  provided the reweighting boosts correct completions relative to incorrect ones.

- **Large sampling budgets ($k \gg 1$):** However, reduced entropy leads to concentration on fewer modes. As β grows, π_θ may collapse onto a narrow subset of correct completions, neglecting other valid solutions accessible under the more dispersed q. Thus,

  $$\limsup_{k\to\infty} \texttt{pass@}k_{\pi_\theta}(x) < \limsup_{k\to\infty} \texttt{pass@}k_q(x),$$

  under typical conditions of entropy reduction and selective mass shifting.

- **Loss of tail coverage:** In particular, if there exist rare but correct completions that have small mass under q but are further downweighted (or eliminated) by the tilting, then the total mass on correct completions can decrease:

  $$\pi_\theta(C) < q(C), \quad C = \{y : R(x,y) = 1\}.$$

  This restricts the long-run probability of recovering diverse solutions via large k sampling.

Conclusion. This establishes a trade-off: RLVR improves sampling efficiency by concentrating probability on high-reward outputs (increasing `pass@1`), but this comes at the cost of reduced entropy and narrower exploration of the solution space (potentially lowering `pass@k` for large k). Empirical studies confirm this phenomenon in settings like code generation and symbolic reasoning, where many semantically distinct correct completions exist.

■

A.7. Estimating the Sampling Threshold ϵ from `pass@k`

We provide a statistical analysis of the threshold ϵ in the `pass@k` sampling. Suppose we sample k times from a model $\pi(\cdot \mid x)$, and let $y^* \in \mathcal{C}$ be a correct completion with unknown probability $p = \pi(y^* \mid x)$. If y^* is not observed in any of those k samples, we can upper bound p using the following argument.

The probability of *not* sampling y^* in a single trial is $1 - p$, so the probability of missing it in all k independent trials is $(1 - p)^k$. To ensure this event occurs with probability at most δ, we solve:

$$(1 - p)^k \leq \delta.$$

Taking logarithms of both sides:

$$k \cdot \log(1 - p) \leq \log \delta.$$

Using the inequality $\log(1 - p) \leq -p$ for $p \in (0, 1)$, we get:

$$k \cdot (-p) \geq \log \delta \quad \Rightarrow \quad p \leq \frac{-\log \delta}{k}.$$

Consequently, if the correct completion y^* is not observed in k samples, then with confidence $1 - \delta$, its probability satisfies:

$$\boxed{\pi(y^* \mid x) \leq \frac{-\log \delta}{k}.}$$

Example. If $k = 8096$ in the math reasoning tasks and we desire 95% confidence (i.e., $\delta = 0.05$), then

$$\pi(y^* \mid x) \leq \frac{-\log(0.05)}{8096} \approx \frac{2.996}{8096} \approx 3.70 \times 10^{-4}.$$

B. Experimental Details

We provide comprehensive details of the experimental setup, including dataset descriptions and evaluation methodologies. A key aspect of our evaluation approach is the answer processing enhancement framework for Reasoning Gym, which addresses format compatibility challenges between base and ProRL models to ensure fair evaluation.

B.1. Evaluation Settings

We employed vLLM (Kwon et al., 2023) as the inference backend. For all models, we utilized a sampling temperature of 0.6, a top_p value of 0.95, and a maximum response length of $32k$.

B.2. Datasets

Math benchmarks. We utilized the complete datasets from MATH500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME 2024, AIME 2025, and AMC 2023 for evaluation.

Non-math benchmarks. For SimpleQA (Wei et al., 2024), we uniformly sampled 10% of the original dataset (433 samples) to enable efficient large-scale evaluation under high-pass conditions. For LiveBench (White et al., 2025), we used the 2024-11-25 version available on HuggingFace. To ensure unambiguous evaluation, we focused exclusively on tasks with binary correct/incorrect judgments and excluded tasks involving intermediate floating-point judgments, as these lack clear correctness criteria. Based on this selection criterion, we evaluated the following subsets: $web\_of\_lies\_v2$ and $spatial$ subsets for Reasoning tasks (LiveBench-R), the $typos$ subset for Language tasks (LiveBench-L), and all available data for Coding tasks (LiveBench-C). For SciBench (Wang et al., 2023), we evaluated on the complete dataset.

Reasoning Gym. For Reasoning Gym (Stojanovski et al., 2025), we employ the `easy` set from the version updated after commit *17a8431* in its repository as our `default` task configuration. This choice ensures consistency with the `default` task configuration used in prior evaluations, maintaining comparable experimental conditions. Additionally, we utilize the `hard` set as our challenging evaluation benchmark for further evaluations.

B.3. Answer Processing Enhancement in Reasoning Gym

We identified significant evaluation challenges when testing the base model on Reasoning Gym. The ProRL model, having been trained on Reasoning Gym data, predominantly produces responses that conform to the expected format, leading to much higher accuracy scores. In contrast, the base model struggled with format adherence due to insufficiently detailed prompts, and its limited 1.5B parameter capacity made it particularly susceptible to evaluation inconsistencies. To address these issues, we enhanced both the answer extraction protocol and prompt design to ensure fair and objective accuracy assessments across both models.

B.3.1. General Answer Extraction Protocol

First, we enhanced the answer extraction protocol with a hierarchical, priority-based extraction mechanism that processes responses through multiple fallback levels. Each level attempts to capture the model's intended answer, and successful extraction at any level bypasses subsequent processing steps.

The strategy first attempts to extract content using the Reasoning Gym's `extract_answer()` function, which captures answers within `<answer></answer>` tags. This approach receives the highest priority since these tags represent Reasoning Gym's default format. When this method fails, the system searches for content within the final `\boxed{}` formatting.

For dice tasks using the base model, failed `extract_answer()` attempts trigger additional processing through Lighteval (Habib et al., 2023)'s `math_normalizer()` function. This function handles `\boxed{}` capture and converts a/b fractions to LaTeX format `\frac{a}{b}`. When `extract_answer()` successfully captures a/b fraction answers, the system applies Lighteval's `fix_a_slash_b()` function to achieve the same LaTeX conversion.

For non-dice tasks or when using ProRL models, failed `extract_answer()` attempts utilize Lighteval's `last_boxed_only_string()` and `remove_boxed()` functions. These functions locate content within the final `\boxed{}`, primarily addressing cases where base model prompt modifications shifted from answer tags to boxed formatting.

As a final fallback, the system extracts content following `</think>` tags when all previous methods fail and the response contains these markers. This safety mechanism captures base model responses that ignore formatting requirements in lengthy tasks.

B.3.2. Task-Specific Processing Modifications

Our core answer processing pipeline applies to both models, with additional processing steps designed primarily to address format compatibility issues commonly encountered with base model responses. Specifically, the processing logic for each task is enhanced as follows:

dice The ground truth for dice tasks uses a/b fraction format. Base models frequently express fractions in LaTeX format, requiring format standardization for accurate evaluation. For base models only, we convert ground truth fractions from a/b format to LaTeX format `\frac{a}{b}` to ensure both model answers and ground truth use consistent LaTeX formatting. ProRL dice processing maintains a/b formatting for both model answers and ground truth, leveraging the dice samples present in its training data.

prime_factorization The ground truth format requires answers to be combinations of numbers and multiplication symbol (i.e., $\times$) only. We implement three key modifications to ensure compatibility with this requirement. First, we standardize LaTeX multiplication symbols by replacing `\times` with $\times$ to meet the evaluation requirements, as base models frequently use LaTeX multiplication symbols instead of standard multiplication signs. Second, we expand LaTeX exponentiation by converting formats like `a^b` into repeated multiplication ($a \times a \times \ldots \times a$ for b iterations), preventing errors when base models consolidate repeated factors into exponential notation. Third, we process response equations by retaining only right-side content when answers contain equals signs, transforming responses like "561 = 3 $\times$ 11 $\times$ 17" to "3 $\times$ 11 $\times$ 17" to eliminate question restatement that base models commonly include.

palindrome_generation The ground truth format expects palindromic character strings (sequences that read the same forwards and backwards). We remove excess whitespace to address frequent spacing issues in base model responses. This

transformation converts spaced responses like "k h g a g h k" to "khgaghk", preventing string reversibility judgment failures that occur when spaces interfere with palindrome verification.

advanced_geometry The ground truth format requires floating-point numbers. Our processing includes three main steps to handle LaTeX formatting issues commonly produced by base models. First, we remove redundant LaTeX expressions by eliminating `\left` and `\right` markers while converting `^\circ` to ° symbol, addressing base models' tendency to use LaTeX for brackets and degree symbols. Second, we convert LaTeX numerical expressions, transforming fractions `\frac{a}{b}` and other LaTeX formats (`\sqrt{a}`, `\sin{a}`, `\log{a}`, `\pi`, etc.) into three-decimal floating-point numbers using the `latex2sympy2_extended` library's `latex2sympy()` function. Third, we evaluate arithmetic expressions containing radicals (such as $2\sqrt{16}+5\sqrt{4}-3$) by converting them into three-decimal floating-point numbers using Python's built-in mathematical functions, handling cases where base models output final results as arithmetic expressions rather than computed values.

power_function The ground truth format uses e-notation scientific notation. We convert mixed LaTeX and arithmetic symbol scientific notation to ensure format consistency. The system transforms patterns like "-2.36×10^{-16}" or "1.5×10^{5}" to e-notation format ("-2.36e-16", "1.5e5"), preventing numerically correct but format-incompatible evaluation errors when base models use mixed LaTeX and arithmetic symbols for scientific notation.

arc_1d The ground truth format requires space-separated digit sequences. We handle two types of responses to meet this grid format requirement. For pure numerical responses, we insert spaces between consecutive digits, converting sequences like "22220000000000000000111" to "2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1". For mixed numerical and textual responses, we extract digits and insert spaces, transforming LaTeX grid formats like `\begin{array}{cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 3 & 0 & 0 & 4 & 6 \\\end{array}` to "0 0 0 0 0 0 0 0 7 3 0 0 4 6", addressing base models' tendency to output correct answers in LaTeX grid format.

boxnet The ground truth format requires dictionary list formatting `[{key: value}, ...]`. We implement comprehensive JSON format cleaning to meet these evaluation requirements. Our processing pipeline includes several steps: rejecting pure numerical responses to prevent non-JSON format interference; removing JSON markdown wrappers that eliminate ` ```json {content} ``` ` markers; converting single dictionaries to single-element dictionary lists (`dict` → `[dict]`); and filtering illegal elements by removing non-dictionary components from JSON lists. Additionally, we clean nested structure values within individual dictionary entries. For nested lists, we extract the first element as the value (`[{key1: [value1, value2, ...]}, ...]` → `[{key1: value1}, ...]`). For nested dictionaries, we select matching key values when available (`[{key1: {key1: value1, key2: value2, ...}}, ...]` → `[{key1: value1}, ...]`) or default to the first element value when keys don't match (`[{key1: {key2: value2, key3: value3}}, ...]` → `[{key1: value2}, ...]`). These modifications preserve model response content to the maximum extent while ensuring ground truth format compliance.

B.4. Entropy Analysis

Setup In entropy analysis, we configure the models with a sampling temperature of 0.6, a top_p value of 0.95, and a maximum response length of $32k$ tokens to balance response diversity and quality. Each model generates 32 completions per problem following the `avg@32` evaluation protocol, and all reported metrics (accuracy, response length, token-level entropy, and answer-level entropy) are averaged across these 32 completions and then across all test problems in each benchmark.

Models We evaluate a diverse set of reasoning models to understand the entropy characteristics across different training paradigms and parameter scales, as summarized in the following table.

Entropy Computation For token-level entropy computation, we employ teacher-forcing to obtain probability estimates. Specifically, after generating the 32 completions with the specified sampling parameters, we feed each generated sequence back to the model and perform a single forward pass to compute the probability distribution over the vocabulary at each token position. Answer-level entropy is computed by first extracting the final answer from each completion using Lighteval (Habib et al., 2023), then calculating the entropy over the distribution of unique answers across the 32 completions. This approach allows us to compute both token-level and answer-level entropy directly from the model's probability distributions without introducing additional sampling variance.

Table 5. Models evaluated in the entropy analysis.

Name	Full Model Name	Type	Parameters
DeepSeek-1.5B	`DeepSeek-R1-Distill-Qwen-1.5B`	Base	1.5B
ProRL-1.5B	`Nemotron-Research-Reasoning-Qwen-1.5B`	RLVR	1.5B
DeepSeek-7B	`DeepSeek-R1-Distill-Qwen-7B`	Base	7B
AceReason-7B	`AceReason-Nemotron-7B`	RLVR	7B
DeepSeek-14B	`DeepSeek-R1-Distill-Qwen-14B`	Base	14B
AceReason-14B	`AceReason-Nemotron-14B`	RLVR	14B
Qwen2.5-32B	`Qwen2.5-32B`	Base	32B
DAPO-32B	`DAPO-Qwen-32B`	RLVR	32B

C. Practical Algorithmic Patterns Explained by RLVR Theory

Recent methods in RLVR often utilize data-filtering strategies and self-supervised reward construction techniques to enhance training stability and reasoning capabilities. While empirically motivated, these techniques can be well understood through our RLVR theoretical framework.

C.1. Prompt Filtering and Selection Heuristics

Several methods (Bae et al., 2025; Zhu et al., 2025) incorporate prompt selection and filtering heuristics to enhance training efficiency. A common strategy is to dynamically filter or down-sample prompts that yield only incorrect completions (`ACC = 0`), thereby avoiding the instability and inefficiency these introduce. For example, Reinforce-Rej (Xiong et al., 2025) reported that retaining such prompts led to high gradient variance and degraded KL efficiency. PODS (Xu et al., 2025) advances this by actively sampling prompts with the highest reward variance to foster strong contrastive signals. Techniques like DAPO's Clip-Higher ensure each mini-batch contains a balanced mix of correct and incorrect completions (`0 < ACC < 1`), sustaining reward variance and diversity. Meanwhile, Polaris (An et al., 2025) and SRPO (Zhang et al., 2025b) further refine selection by excluding trivial prompts (`ACC = 1`) that offer little gradient information. Though initially inspired by GRPO, which implicitly nullifies gradients on all-wrong prompts, these designs closely reflect our theoretical findings: Thm.2.2 shows zero-accuracy prompts provide no useful signal, Thm.2.8 warns that over-emphasizing easy prompts risks entropy collapse and diminished `pass@k`, and Prop. 2.6 underscores that meaningful updates rely on in-support reward variability.

C.2. Self-Supervised Bootstrap Learning

Recent studies (Prabhudesai et al., 2025; Shao et al., 2025) explore self-supervised RL techniques that improve reasoning without external ground-truth labels. Unlike traditional RLVR relying on externally verifiable rewards, they build intrinsic reward signals from the model's outputs, leveraging internal consistency, semantic coherence, or self-play. For example, EMPO (Zhang et al., 2025a) minimizes semantic entropy across self-generated clusters, RLIF (Zhao et al., 2025c) uses model confidence scores (INTUITOR) as intrinsic rewards, SRT (Shafayat et al., 2025) and TTRL (Zuo et al., 2025) employ majority voting among completions, Absolute Zero (Zhao et al., 2025a) adopts a self-play mechanism guided by environment feedback, and EM-RL (Agarwal et al., 2025) combines entropy regularization with gradient alignment. Even random intrinsic rewards can yield improvements by exploiting model inductive biases (Shao et al., 2025). These approaches succeed by aligning rewards with natural structural redundancies in model outputs. EMPO and RLIF favor internally coherent or confident completions, SRT and TTRL reinforce consensus, Absolute Zero evolves proposal and solution generations through self-play, and EM-RL maintains stability by regulating entropy and gradients. Viewed through our lens, they implicitly respect theoretical constraints: consistent with Thm. 2.2, they remain within the base model's support through sampling or clustering; aligned with Thm. 2.8, they systematically reduce entropy, sharpening distributions over promising modes; and as shown by Prop. 2.6, they maintain stability via entropy-regularized or gradient-constrained updates.