

New Skills or Sharper Primitives? A Probabilistic Perspective on the Emergence of Reasoning in RLVR

Zhilin Wang^{1,2} Yafu Li^{2,3*} Shunkai Zhang⁴ Zhi Wang⁵
Haoran Zhang^{2,6} Xiaoye Qu² Yu Cheng^{3*}

¹University of Science and Technology of China ²Shanghai AI Laboratory

³The Chinese University of Hong Kong ⁴Peking University

⁵Nanjing University ⁶Shanghai Jiao Tong University

zhilin.nlp@gmail.com yafuly@gmail.com chengyu@cse.cuhk.edu.hk

Abstract

Whether Reinforcement Learning with Verifiable Rewards (RLVR) creates new reasoning capability or mainly sharpens latent behavior remains debated. We develop a probabilistic account of how RLVR can create composite capability from sharpened atomic transitions. First, we model capability as instance-level solvability $P_\theta(q)$ under a finite sampling budget and use this view to reinterpret dataset-level $Pass@k$. Second, we analyze how RLVR can make composite capability appear by sharpening weak atomic transitions and pushing joint success probability across a multiplicative barrier. Third, in Algebrarium, we train only on 1-hop atomic operations and evaluate on unseen 2–5-hop compositions. In this setting, RLVR moves 22.6% of base-null model–task pairs beyond the finite-budget feasibility threshold, and composite success is strongly predicted by the joint probability of atomic steps (Pearson $\rho \in [0.69, 0.96]$). These results show that some seemingly new capabilities in RLVR arise as the finite-budget manifestation of sharpened atomic transitions.

1 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) (Shao et al., 2024) is now a central recipe for improving LLM reasoning (OpenAI et al., 2019; DeepMind, 2025). Yet a basic question remains unresolved: does RLVR create genuinely new capabilities, or does it mainly concentrate behavior already weakly present in the base model? Recent work argues for both views. Some papers report composite skills that appear only after RLVR (Yuan et al., 2025; Cheng et al., 2025); others argue that RLVR largely exploits traces already available in the pretrained policy (Yue et al., 2025; Gandhi et al., 2025; Liu et al., 2025; Zhao et al., 2025). This paper studies that question through a probabilistic lens.

Much of this debate has been framed through aggregate $Pass@k$: improvements in $Pass@k$ are often taken as evidence that RLVR has induced new capability, while degradation at high k is taken as evidence that RLVR merely concentrates existing behavior (Yue et al., 2025; Wen et al., 2025). This is incomplete. Because $Pass@k$ is a dataset-level macro-statistic, it does not reveal how solvability changes on individual instances. We therefore study a concrete question: can RLVR induce *new instance-level capability under a finite sampling budget* by sharpening low-probability atomic transitions and making previously unsampleable multi-step solutions reachable?

The intuition is simple. Complex reasoning often requires several intermediate decisions to be correct in sequence (Wei et al., 2023; Dziri et al., 2023). A model may already solve many atomic subproblems with moderate probability, yet still fail on the full task because the joint success probability collapses across the chain. If RLVR reliably raises the probability of key atomic steps, the compounded effect can move a composite task from effectively impossible to feasible under a finite sampling budget.

*Corresponding authors

To formalize this idea, we analyze capability through instance-level solvability $P_\theta(q)$ rather than dataset-level averages alone. This lets us reinterpret $Pass@k$ as a macro-statistic induced by per-instance success probabilities. The $Pass@k$ identity itself is classical; our contribution is to use it to explain why RLVR can produce very different macro-level $Pass@k$ patterns depending on how it redistributes probability mass across instances.

We test this account in Algebrarium, a symbolic environment with exact verifiability and explicit atomic/composite structure. We train models only on atomic operations and evaluate them on unseen multi-step compositions. Because the atomic/composite split is defined relative to the target symbolic environment, our setup does not assume that pretraining globally contains only single-hop knowledge.

Our contributions are threefold:

- **A probabilistic interpretation of $Pass@k$.** We decompose dataset-level $Pass@k$ into instance-level solvability and use this view to explain why low- k and high- k behavior can diverge after RLVR.
- **A probabilistic mechanism for RLVR-induced emergence.** We define finite-budget capability at the instance level and argue that RLVR can make seemingly new composite capability appear by sharpening weak atomic transitions and pushing joint success probability across the multiplicative barrier.
- **Controlled empirical support.** In an atomic-only RLVR setup, 22.6% of pooled model-task pairs with 0/128 correct base samples meet our strict post-RLVR feasibility criterion, and composite success is strongly predicted by the joint probability of atomic steps (Pearson $\rho \in [0.69, 0.96]$).

Together, these results link our capability definition, our probabilistic analysis of $Pass@k$, and our atomic-to-composite experiments. Under this perspective, seemingly new capabilities induced by RLVR emerge as the finite-budget manifestation of sharpened atomic transitions.

2 Related Work

RLVR and the emergence debate. RLVR is now a standard paradigm for improving reasoning performance in LLMs (OpenAI, 2024; DeepMind, 2025). These methods reinforce sampled trajectories that receive higher verifiable reward than their peers (DeepSeek-AI et al., 2025; Shao et al., 2024; Yu et al., 2025; Zheng et al., 2025). Their success has triggered a core debate: does RLVR mainly elicit behavior already present in the base policy, or can it create genuinely new capability under practical inference budgets? (Yue et al., 2025)

RLVR as elicitation. A growing literature argues that RLVR mainly amplifies pre-existing competence rather than injecting new skill (Zhao et al., 2025; Wen et al., 2025). Empirically, RLVR is often associated with entropy reduction (Cui et al., 2025; Wang et al., 2025; Jiang et al., 2025; Yan et al., 2025) and, in some settings, degraded high- k $Pass@k$ despite stronger greedy performance (Yue et al., 2025; Zhu et al., 2025; He et al., 2025). In this view, RLVR depends on spontaneous sampling of correct trajectories and therefore cannot reward behaviors absent from the model’s reachable distribution (Zhan et al., 2026).

RLVR as compositional induction. Other work reports composite skills that appear after RLVR, especially in controlled or synthetic settings that separate atomic operations from compositional reasoning (Yuan et al., 2025; Cheng et al., 2025; Park et al., 2025). These studies suggest that RLVR can produce behavior that is meaningfully new at evaluation time, even if it is built from weaker pre-existing components. Our work is closest to this line, but focuses more narrowly on one measurable mechanism in a controlled setting.

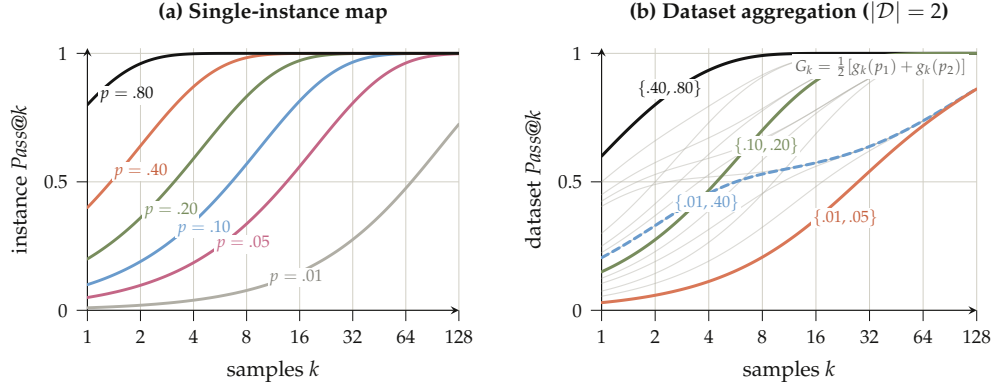


Figure 1: **From instance solvability to dataset-level $Pass@k$.** (a) For a single instance, the success probability p determines $g_k(p) = 1 - (1 - p)^k$, shown for $p \in \{.01, .05, .10, .20, .40, .80\}$. (b) For a two-instance dataset, $Pass@k$ is the arithmetic mean of the two instance curves. Faint lines show all 15 pairwise compositions, while four representative pairs are highlighted. Different pairs can produce similar or crossing aggregate trajectories, so the macro curve does not identify which instances changed.

3 Deconstructing Aggregate $Pass@k$

3.1 Why Aggregate $Pass@k$ Is Insufficient

$Pass@k$ is a standard finite-sampling metric (Chen et al., 2021). We use its classical identity to show why aggregate $Pass@k$ is insufficient for the RLVR emergence debate. The identity below connects dataset-level $Pass@k$ to the distribution of instance-level solvability $P_\theta(q)$, which is the object needed for our analysis.

Given a dataset $\mathcal{D} = \{q_i\}_{i=1}^N$, $Pass@k$ measures the proportion of problems solved by at least one correct sample among k generations:

$$Pass@k = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathbb{I}(\text{solved}_i \geq 1), \quad (1)$$

where $\text{solved}_i \in \{0, 1, \dots, k\}$ is the number of correct samples for problem q_i .

Let $\pi_\theta(y | q)$ denote the policy and let $V(y, q) \in \{0, 1\}$ be the verifier. We define the success probability of a single task as

$$P_\theta(q) = \mathbb{E}_{y \sim \pi_\theta(\cdot | q)} [V(y, q)]. \quad (2)$$

If the k samples are independent, the probability that all of them fail is $(1 - P_\theta(q))^k$, so the expected $Pass@k$ for one instance is

$$\mathbb{E}[Pass@k(q)] = 1 - (1 - P_\theta(q))^k. \quad (3)$$

Averaging over the dataset gives

$$\mathbb{E}[Pass@k] = \frac{1}{N} \sum_{i=1}^N \left[1 - (1 - P_\theta(q_i))^k \right]. \quad (4)$$

This decomposition matters because aggregate $Pass@k$ is only a macro-statistic. By itself, it cannot tell us whether RLVR has created new instance-level capability, merely sharpened already-solvable cases, or traded coverage in one region of the task distribution for gains in another.

Takeaway. Different redistributions of $P_\theta(q)$ across instances can generate similar aggregate $\text{Pass}@k$ curves. Settling the emergence debate therefore requires instance-level analysis, not macro-statistics alone.

3.2 From Theory to Estimation

Figure 1 illustrates how dataset-level curves arise from instance-level success probabilities. This view also clarifies why prior RLVR studies can report different post-training $\text{Pass}@k$ patterns: if training redistributes $P_\theta(q)$ unevenly across instances, low- k can improve while high- k degrades (Yue et al., 2025). The key point is that macro-level $\text{Pass}@k$ curves do not uniquely identify the underlying redistribution.

4 Capability as Probabilistic States

To reason about capability directly, we move from dataset-level averages to pointwise solvability. We treat capability as an operational property of a model-task pair under a finite sampling budget.

We partition the task space according to the magnitude of $P_\theta(q)$ using two thresholds: a lower threshold ϵ for tasks that are effectively unsampleable, and a higher threshold δ for tasks that are meaningfully feasible under the intended compute budget.

Definition 1: The Null Set (ϵ -incapability). A task belongs to the Null Set $\mathcal{N}_\theta^\epsilon$ if its success probability is below an operational lower bound:

$$\mathcal{N}_\theta^\epsilon = \{q \in \mathcal{Q} \mid P_\theta(q) < \epsilon\}. \quad (5)$$

We estimate ϵ using the Rule of Three (Hanley & Lippman-Hand, 1983). If zero successes are observed in $K_{\text{large}} = 128$ samples, a standard 95% upper bound on the true success rate is approximately $3/128 \approx 0.023$. We therefore use $\epsilon \approx 0.023$ as a finite-budget threshold for the null regime. This is an operational boundary, not a universal boundary of capability.

Definition 2: The Feasible Set (δ -capability). A task belongs to the Feasible Set $\mathcal{F}_\theta^\delta$ if its success probability is high enough to make success plausible under the intended inference budget:

$$\mathcal{F}_\theta^\delta = \{q \in \mathcal{Q} \mid P_\theta(q) \geq \delta\}. \quad (6)$$

We set $\delta = 1/K_{\text{min}} = 0.125$ for a reference budget of $K_{\text{min}} = 8$. Under this threshold, the expected number of correct samples is at least one:

$$K_{\text{min}} \cdot P_\theta(q) \geq 1. \quad (7)$$

This does not guarantee success on every run, but it separates reliably reachable tasks from extreme low-probability outliers. Like ϵ , this threshold is operational rather than canonical: it fixes the inference budget relative to which we evaluate finite-budget capability, and different deployment budgets would induce different feasible sets.

The transitional regime. Tasks with $\epsilon \leq P_\theta(q) < \delta$ lie in a stochastic frontier between the two sets. They are neither clearly impossible nor reliably feasible under the chosen budget. In the analyses below, we focus on transitions between the null and feasible regimes because those are the clearest indicators of finite-budget capability change.

5 Compositional Emergence under a Multiplicative Barrier

5.1 The Multiplicative Barrier in Complex Reasoning

Consider a complex problem q decomposed into a sequence of M substeps $\mathcal{S} = \{s_1, s_2, \dots, s_M\}$. The exact autoregressive factorization is

$$P_\theta(q) = \prod_{j=1}^M P_\theta(s_j \mid s_{<j}). \quad (8)$$

The exact object is the conditional chain above. For later quantitative analysis, we also use the first-order proxy

$$P_\theta(q) \approx \prod_{j=1}^M P_\theta(s_j), \quad (9)$$

to summarize stepwise strength.

The basic barrier does not depend on strict independence. If several conditional substeps are weak, their joint probability still collapses rapidly with chain length. For example, if five required steps each succeed with probability around 0.3, then

$$P_\theta(q) \approx 0.3^5 \approx 0.0024,$$

which lies far below the null threshold $\epsilon \approx 0.023$. In that regime, the model may solve each atomic operation with non-trivial probability while remaining effectively unable to solve the full composite task under finite sampling.

5.2 Emergence through Probability Amplification

In our controlled setting, RLVR trains only on atomic tasks. If training sharpens several weak atomic transitions, the compounded joint probability can cross the feasibility boundary:

$$\underbrace{0.3^5 \approx 0.0024}_{\substack{\text{Null regime} \\ (P_\theta(q) < \epsilon)}} \xrightarrow[\text{sharpening atomic steps}]{\text{RLVR}} \underbrace{0.7^5 \approx 0.168}_{\substack{\text{Feasible regime} \\ (P_\theta(q) \geq \delta)}}.$$

This is the mechanism we study: atomic-only RLVR can create new *instance-level* capability under a finite budget by sharpening weak atomic transitions and pushing joint success probability across the feasibility boundary.

6 Experiment Setup

To study the proposed mechanism, we use Algebrarium to generate synthetic symbolic reasoning tasks. This gives a laboratory setting in which atomic operations and composite tasks are separated cleanly, answers are deterministically verifiable, and the reasoning process is measurable at both the atomic and composite levels. Details of the framework appear in Appendix A.

Domains and task formulation. The core task is *algebraic expression calculation*. Given two elements $e_1, e_2 \in G$ from a specific group, the model must compute the canonical result of $e_1 \oplus e_2$ by applying the group rules of that domain. To probe robustness across different algebraic properties, we instantiate four domains spanning two axes: cardinality (finite vs. infinite) and commutativity (abelian vs. non-abelian).

Measurement and decomposition. We quantify composite difficulty by operation count N and compare direct success $P_\theta(q)$ with the success of each decomposed atomic step $P_\theta(s_j)$. This gives three aligned quantities for each instance: final-answer accuracy, per-step atomic accuracy, and a joint atomic proxy $\prod_{j=1}^N P_\theta(s_j)$. Because each atomic step is measured with labeled intermediate targets, this product serves as a diagnostic proxy. The full evaluation schema appears in Appendix 2.

A concrete decomposition example. Suppose the target expression is a 3-hop composition such as $((e_1 \oplus e_2) \oplus e_3) \oplus e_4$. Direct evaluation asks whether the model can generate the final result in one rollout. Decomposed evaluation instead measures the three atomic transitions separately: $s_1 : e_1 \oplus e_2 \rightarrow d_1$, $s_2 : d_1 \oplus e_3 \rightarrow d_2$, and $s_3 : d_2 \oplus e_4 \rightarrow \text{Res}$. This gives a concrete instance-level bridge between the final composite outcome and the intermediate atomic operations on which our analysis is built.

Table 1: **Dataset statistics and domain properties.** We evaluate four algebraic systems varying in cardinality and commutativity. Training contains only atomic examples, while the test set contains 800 composite examples stratified by reasoning depth (2–5 operations).

Domain	Group Properties		Underlying Structure	Train	Test (Stratified by Depth)					Total
	Card.	Comm.			1-Ops	2-Ops	3-Ops	4-Ops	5-Ops	
Encrypted History	∞	Yes	Integers ($\mathbb{Z}$)	3,200	50	50	50	50	50	200
Enigma System	Finite	Yes	Modular Product ($\mathbb{Z}_{26}^3$)	3,200	50	50	50	50	50	200
Knitting System	∞	No	Free Group	3,200	50	50	50	50	50	200
Rubik’s Cube	Finite	No	Permutation Group	3,200	50	50	50	50	50	200
Total	<i>4 Distinct Algebraic Systems</i>			12,800	200	200	200	200	200	800

Dataset construction. Our split is designed to decouple rule acquisition from compositional generalization.

- **Training:** The RLVR training set contains only atomic operations (1-hop reasoning). This rules out direct RL on composite targets.
- **Testing:** The test set contains unseen 2–5-hop compositions, stratified by depth, with 200 balanced samples per level across the four domains.

The atomic/composite split is defined relative to this target environment, not relative to the model’s entire pretraining history. A pretrained LLM may bring generic reasoning priors, but the symbolic rules and held-out compositions are generated specifically for this environment.

Table 1 summarizes the full dataset configuration. Task-specific details are deferred to Appendix C.

Implementation details. We benchmark three model families: LLAMA, QWEN, and GEMMA. Training uses Group Relative Policy Optimization (GRPO) (Shao et al., 2024) for 100 steps. Performance is measured by exact-match accuracy on the final boxed answer. Additional implementation details are provided in Appendix B.

7 Empirical Validation of the Theory

This section tests the probabilistic account from Sections 3.1–5. Aggregate $Pass@k$ improvement alone is not our evidence of emergence. Instead, we ask three questions: whether the instance-level decomposition describes observed $Pass@k$ curves, whether atomic-only RLVR induces finite-budget transitions on unseen composite tasks, and whether composite success tracks atomic-step strength as predicted by the multiplicative barrier.

7.1 Validation of $Pass@k$ Dynamics

We first test whether the classical $Pass@k$ identity provides a useful estimator of empirical curve shape. For each instance, we approximate $P_\theta(q)$ by the average correctness rate over 128 samples, i.e., $P_\theta(q) \approx \text{Avg@128}(q)$ using Eq. 2. We then plug these estimates into the theoretical expression and compare the resulting curves with the measured $Pass@k$ values.

Figure 2 shows close agreement between the predicted and empirical curves, with MSE below 6×10^{-4} across all model–state pairs. This validates the measurement framework: the observed dataset-level $Pass@k$ curves can be reconstructed from the distribution of $P_\theta(q)$. The fit is precise because the reconstruction uses the empirically estimated per-instance success rates directly and introduces no additional free parameters. The point of this figure is therefore methodological: once the instance-level distribution is measured, the macro $Pass@k$ curve is almost fully determined.

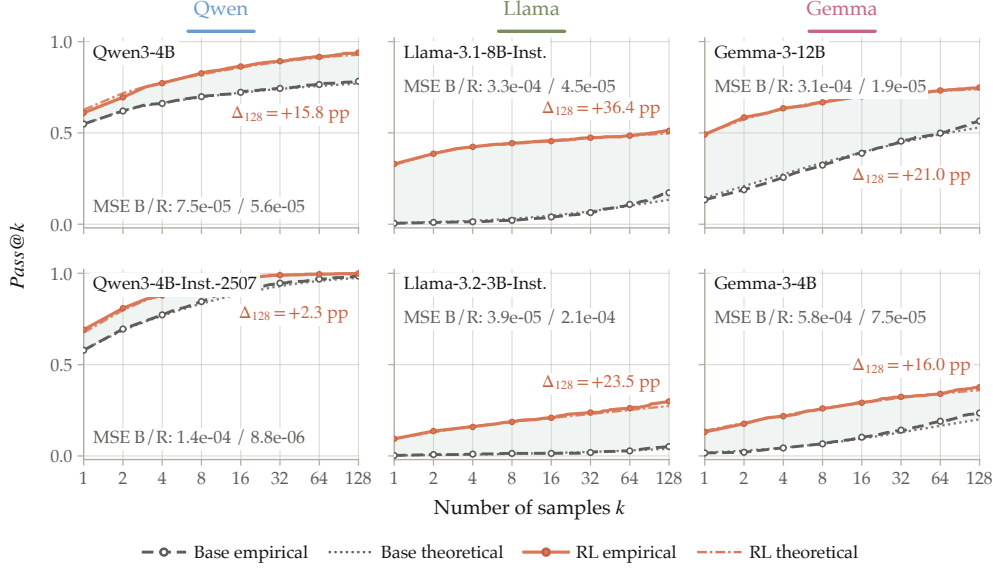


Figure 2: **Theoretical and empirical $Pass@k$ curves.** Using per-instance success rates estimated from 128 samples, the theoretical curves closely track the observed curves (MSE 8.8×10^{-6} – 5.9×10^{-4} across model–state pairs).

7.2 Transitions from the Null Regime

We next ask whether atomic-only RLVR can create new *instance-level* capability on unseen composite tasks under a finite sampling budget. Following Section 4, for each evaluated model we isolate model–task pairs that lie in the Null Set for the base model, defined operationally as zero successes in 128 samples. We then test whether RLVR moves these same pairs beyond the feasibility threshold under the predeclared reference budget $K_{\min} = 8$. To make the empirical transition count conservative at the discrete boundary, we require $\hat{P}_{RL}(q) > \delta$ (at least 17 successes in 128 samples); pairs exactly at δ are not counted as transitions.

As shown in Figure 3, 22.6% of these initially null model–task pairs exceed the feasibility threshold after RLVR, and the mean success rate of the recovered subset rises from 0 to 0.60. The key event here is an instance transition across capability regimes, not merely a shift in an aggregate curve. Because the RLVR training set contains only atomic tasks, these gains show that RLVR can create new finite-budget capability on unseen composite instances without direct composite supervision.

7.3 Testing the Multiplicative Barrier Hypothesis

We now examine whether composite difficulty in this environment is consistent with the proposed multiplicative barrier. We use two complementary tests: performance as a function of reasoning depth, and the relationship between composite success and the joint accuracy of constituent atomic steps.

Depth-dependent collapse. Figure 4 shows that accuracy drops sharply as reasoning depth increases. The overall trend is well approximated by an exponential form $y \approx p^x$, which supports the idea that a small number of weak links can make long chains effectively unsampleable.

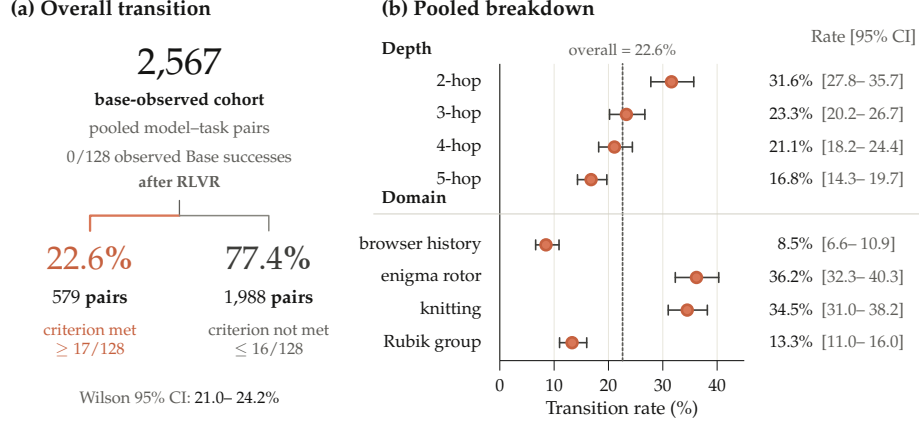


Figure 3: **Transitions from the null regime.** (a) Among 2,567 pooled model-task pairs with 0/128 correct base-model samples, 579 achieve at least 17/128 after RLVR (22.6%; pair-level Wilson 95% CI: 21.0–24.2%). (b) Point estimates show a decreasing trend with reasoning depth and vary across domains; intervals are pair-level Wilson 95% confidence intervals and the dotted line marks the pooled overall rate.

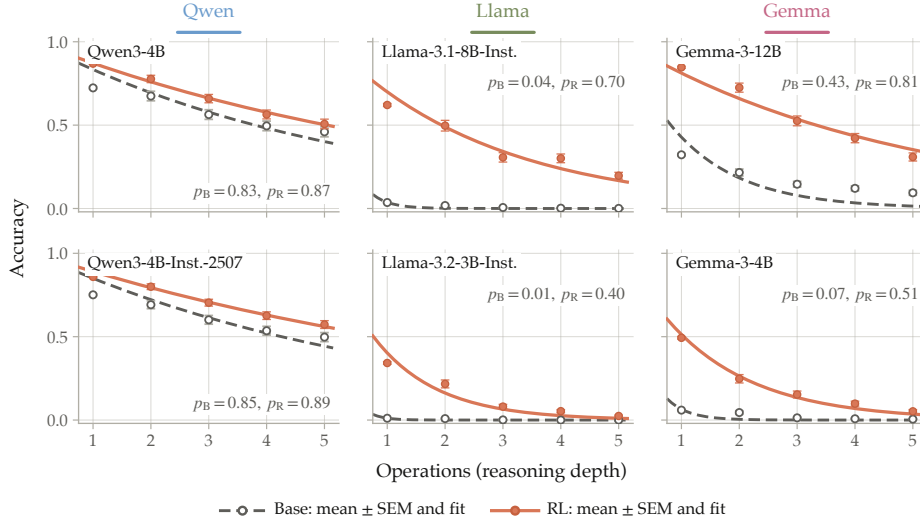


Figure 4: **Accuracy versus reasoning depth.** Composite accuracy falls rapidly as the number of required operations increases, consistent with a multiplicative compounding effect.

Atomic steps versus final outcome. Figure 5 relates final-answer success to the joint atomic proxy $\prod P_\theta(s_j)$. Before RLVR, several base models cluster near the origin, indicating that their atomic chains are too weak for the composite task to appear under finite sampling. After RLVR, the coupling between atomic-step strength and final success becomes substantially tighter, with Pearson correlations in the range $\rho \in [0.69, 0.96]$. The important change is not just that both quantities rise, but that they align more systematically after training: recovered composite tasks concentrate where the underlying atomic chain has also been strengthened. Because the proxy is computed from independently measured, teacher-forced atomic steps rather than from free composite rollouts, it serves as a diagnostic view of the barrier. Even so, the tighter relationship shows that composite gains track sharpened atomic transitions.

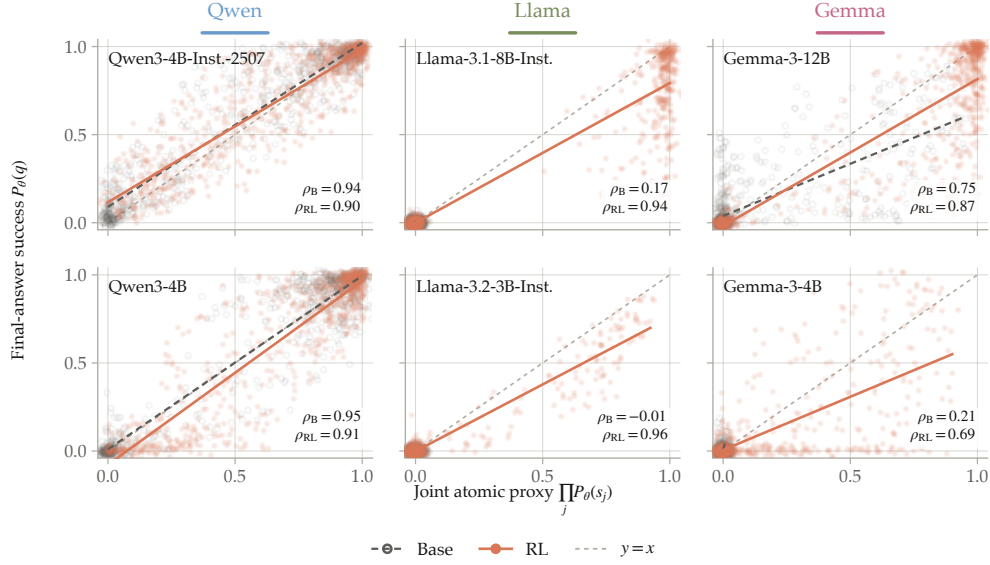


Figure 5: **Atomic-step strength versus composite success.** The plots compare the joint atomic proxy $\prod_j P_\theta(s_j)$ with final-answer success $P_\theta(q)$. Weaker base models cluster near the origin, while RL-tuned models show a tighter positive relationship.

7.4 Why Atomic RLVR Polarizes Composite $Pass@k$

Figure 6 shows that atomic-only RLVR redistributes skill accuracy. Gains in weak bottlenecks can make composite chains reachable and raise low- k $Pass@k$, while regressions in reliable steps erode longer chains and lower high- k $Pass@k$. The trade-off is stronger for Llama and Gemma than for Qwen.

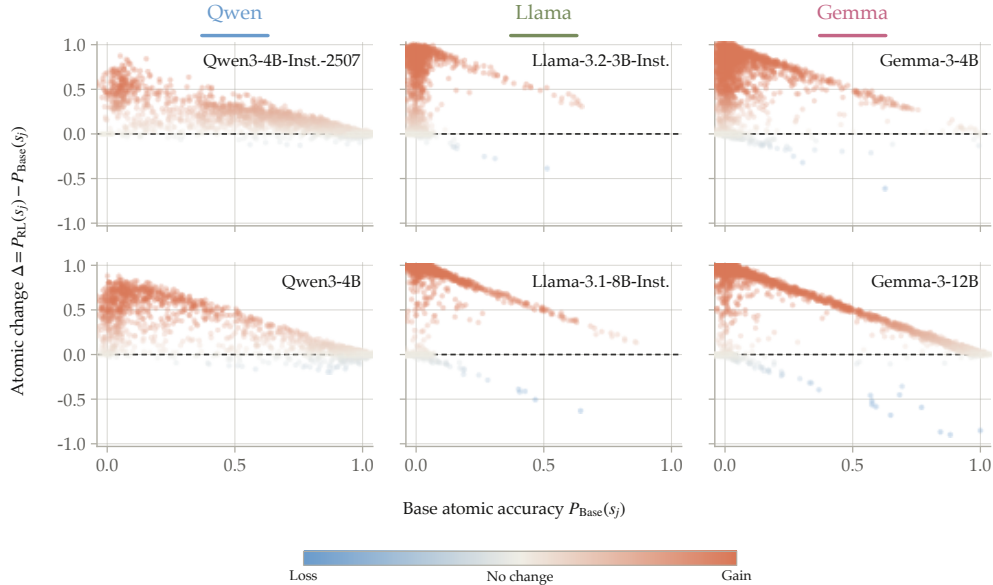


Figure 6: **Atomic performance shifts after RLVR.** Gains in weak skills and regressions in strong skills help explain opposite low- and high- k effects on composite $Pass@k$.

8 Conclusion

We present a probabilistic perspective on the RLVR emergence debate. Viewing capability as instance-level solvability under a finite inference budget clarifies why aggregate *Pass@k* cannot by itself determine whether new capability has appeared. In the atomic-only Algebrarium setting, RLVR moves a non-trivial subset of unseen composite tasks from the null regime to the feasible regime, and these gains track the strength of constituent atomic steps.

In this setting, some seemingly new capabilities induced by RLVR arise as the finite-budget manifestation of sharpened atomic transitions. A secondary result is that atomic RLVR can polarize composite *Pass@k*: strengthening bottleneck steps raises low-*k*, while weakening other supporting steps can lower high-*k* on complex chains. Together, these results provide a measurable mechanism and an instance-level framework for studying RLVR emergence.

References

- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021.
- Sitao Cheng, Xunjian Yin, Ruiwen Zhou, Yuxuan Li, Xinyi Wang, Liangming Pan, William Yang Wang, and Victor Zhong. From atomic to composite: Reinforcement learning enables generalization in complementary reasoning, 2025. URL <https://arxiv.org/abs/2512.01970>.
- Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL <https://arxiv.org/abs/2505.22617>.
- Google DeepMind. Gemini 2.5: Pushing the frontier with advanced reasoning, multi-modality, long context, and next generation agentic capabilities. *ArXiv*, abs/2507.06261, 2025.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bing-Li Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Jiong Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, M. Tang, Meng Li, Miaojuan Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, Ruiqi Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shao-Kang Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen-Xia Yu, Wentao Zhang, Wangding Xiao, Wei An, Xiaodong Liu,

- Xiaohan Wang, Xi aokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyu Jin, Xi-Cheng Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yu-Jing Zou, Yujia He, Yunfan Xiong, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yao Li, Yi Zheng, Yuchen Zhu, Yunxiang Ma, Ying Tang, Yukun Zha, Yuting Yan, Zehui Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zi-An Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Nature*, 645, 2025.
- Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality, 2023. URL <https://arxiv.org/abs/2305.18654>.
- Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, nathan lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *COLM*, 2025.
- James A Hanley and Abby Lippman-Hand. If nothing goes wrong, is everything all right?: interpreting zero numerators. *Jama*, 249(13):1743–1745, 1983.
- Andre He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*, 2025.
- Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. Rethinking entropy regularization in large reasoning models, 2025. URL <https://arxiv.org/abs/2509.25133>.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *COLM*, 2025.
- OpenAI. Openai o1 system card. *ArXiv*, 2024.
- OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik’s cube with a robot hand. *CoRR*, abs/1910.07113, 2019. URL <http://arxiv.org/abs/1910.07113>.
- Simon Park, Simran Kaur, and Sanjeev Arora. How does rl post-training induce skill composition? a case study on countdown, 2025. URL <https://arxiv.org/abs/2512.01775>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300, 2024.
- Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL <https://arxiv.org/abs/2201.11903>.

- Xumeng Wen, Zihan Liu, Shun Zheng, Zhijian Xu, Shengyu Ye, Zhirong Wu, Xiao Liang, Yang Wang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.
- Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance, 2025. URL <https://arxiv.org/abs/2504.14945>.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL <https://arxiv.org/abs/2503.14476>.
- Lifan Yuan, Weize Chen, Yuchen Zhang, Ganqu Cui, Hanbin Wang, Ziming You, Ning Ding, Zhiyuan Liu, Maosong Sun, and Hao Peng. From $f(x)$ and $g(x)$ to $f(g(x))$: Llm learn new skills in rl by composing old ones, 2025. URL <https://arxiv.org/abs/2509.25123>.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *ArXiv*, 2025.
- Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F. Wong, and Yu Cheng. ExGRPO: Learning to reason from experience. In *International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=701tjQXWVWk>.
- Rosie Zhao, Alexandru Meterez, Sham M. Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: RL post-training amplifies behaviors learned in pretraining. *CoRR*, abs/2504.07912, 2025. doi: 10.48550/ARXIV.2504.07912. URL <https://doi.org/10.48550/arXiv.2504.07912>.
- Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL <https://arxiv.org/abs/2507.18071>.
- Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025.

A Algebrarium Framework

To evaluate symbolic reasoning in a controlled setting while reducing contamination risk, we developed **Algebrarium**. Unlike static benchmarks, Algebrarium procedurally generates algebraic tasks from user-defined mathematical structures. This makes atomic/composite structure explicit and keeps answers exactly verifiable.

The core design principle of Algebrarium is to decouple *abstract mathematical definition* from *concrete task generation*. This allows many reasoning problems to be derived from the same logical core while keeping generation analyzable. The framework follows three principles:

1. Declarative Algebraic Structures Algebrarium treats algebra as a formal system defined by a tuple $(S, \Sigma, \mathcal{A})$, where S denotes a carrier set, Σ a signature of operations, and $\mathcal{A}$ the governing rules. These structures are defined declaratively, which lets the framework generate both standard systems, such as modular arithmetic, and non-standard symbolic systems tailored for controlled evaluation.

2. Procedural Task Construction Rather than retrieving pre-existing questions, the framework constructs reasoning tasks algorithmically. A generation engine instantiates algebraic instances and produces symbolic artifacts, such as expression trees or equations, from explicit rules. This pipeline includes:

- **Controllable Complexity:** Parameters including recursion depth, expression length, and the cardinality of the underlying set can be adjusted to create a smooth difficulty gradient.
- **Diverse Task Types:** The engine supports various symbolic tasks, ranging from forward evaluation, such as reducing an expression $a \circ b \circ c$ to a single element, to inverse reasoning, which involves solving equations like $a \circ x = b$ or systems of group equations.

Table 2: **Evaluation framework.** We distinguish direct evaluation on the full task from evaluation on decomposed binary operations. Intermediate results d_j are treated as labeled targets, which allows atomic-step measurement independent of final-answer measurement. The product $\prod_{j=1}^N P_\theta(s_j)$ is used as a diagnostic proxy rather than as an exact conditional chain probability.

Mode	Structure (Binary Ops)	Metric
Direct	$q \xrightarrow{\text{Infer}} \text{Result}$	$P_\theta(q)$
Decomposed	<i>Sequential Chain</i>	
Step 1	$s_1 : e_1 \oplus e_2 \rightarrow d_1$	$P_\theta(s_1)$
...	...	...
Step j	$s_j : e_k \oplus d_{j-1} \rightarrow d_j$	$P_\theta(s_j)$
Step N	$s_N : e_{\text{final}} \oplus d_{N-1} \rightarrow \text{Res}$	$P_\theta(s_N)$
Definitions	• Atomic: $P_\theta(s_j)$ • Result: $P_\theta(q)$ • Joint proxy: $\prod_{j=1}^N P_\theta(s_j)$	

3. Verifiable Ground Truth Because the data is generated from formal rules, the ground truth is deterministic and mathematically verifiable. This supports exact-match evaluation and, when needed, automatic generation of step-by-step traces.

A.1 Evaluation Schema

The main text compares two views of each composite problem: direct solvability of the full task and stepwise solvability of its atomic decomposition. Table 2 summarizes this evaluation schema. Direct evaluation measures whether the model can solve the whole expression in one rollout, while decomposed evaluation measures the success rate of each labeled atomic transition. This yields the three quantities used throughout the paper: final-answer success $P_\theta(q)$, per-step atomic success $P_\theta(s_j)$, and the joint atomic proxy $\prod_{j=1}^N P_\theta(s_j)$.

B Implementation Details and Training Dynamics

B.1 Base Models and Infrastructure

We evaluate six models from three families:

- **Qwen3:** Qwen3-4B and Qwen3-4B-Instruct.
- **Llama-3.1:** Llama-3.1-8B-Instruct and Llama-3.1-3B-Instruct.
- **Gemma-3:** gemma-3-4b-it and gemma-3-12b-it.

All models are trained with the ver1 framework and use vLLM for rollout generation. We use Tensor Parallelism ($TP = 2$) and allow responses up to 8,192 tokens.

B.2 Hyperparameter Configuration

For all experiments, we train with Group Relative Policy Optimization (GRPO) and AdamW at a learning rate of 1×10^{-6} for 100 optimization steps. The global batch size is 128, the mini-batch size is 8, and the group size is $G = 8$, with gradient accumulation enabled. Maximum prompt and response lengths are 1,024 and 8,192 tokens. We monitor KL divergence with penalty coefficient $\beta = 0.001$, but do not include the KL term in the optimization objective. Advantage normalization is disabled.

B.3 Training Dynamics

Figure 7 summarizes the 100 optimization steps across model families: critic reward rises rapidly, while actor entropy falls and then stabilizes.

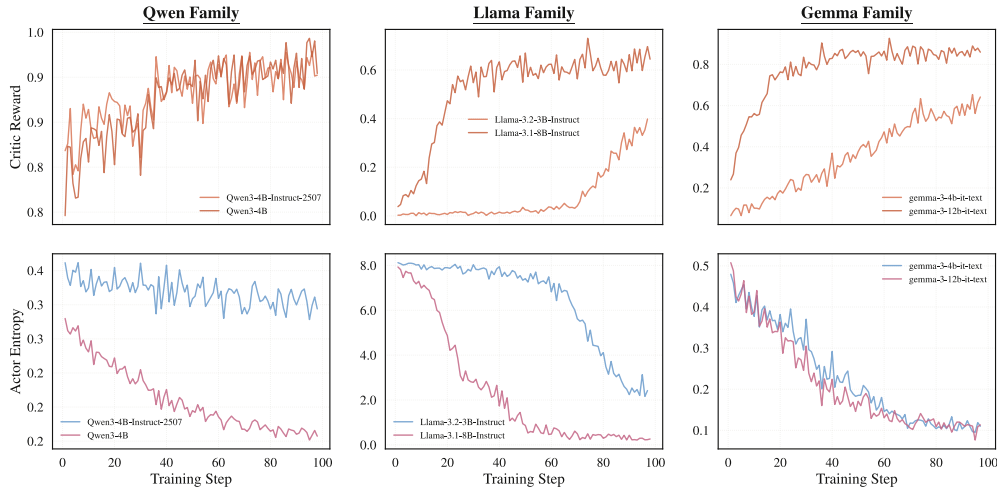


Figure 7: **Training dynamics across model families.** Critic reward rises while actor entropy falls and stabilizes across Qwen, Llama, and Gemma.

C Domain Specifications and Operational Semantics

This section gives the formal specifications of the four algebraic systems used in our experiments. Together they cover the four combinations of finite vs. infinite and abelian vs. non-abelian structure.

C.1 Encrypted History Navigation (Infinite Abelian Group)

This domain models 1-dimensional navigation on an integer line, obfuscated by a substitution cipher. It tests the model’s ability to perform arithmetic operations ($\mathbb{Z}$) under symbolic encryption.

Representation

- **Base System:** Base-7.
- **Cipher Alphabet:** A bijective mapping $\phi : \{0, \dots, 6\} \rightarrow \{a, \dots, g\}$ where $0 \mapsto a, 1 \mapsto b, \dots, 6 \mapsto g$.
- **Structure:** Elements are strings in the format $\text{DIR}(\text{val})$, where $\text{DIR} \in \{\text{FWD}, \text{BACK}\}$ and val is a string of cipher characters representing a Base-7 magnitude.

Operational Logic Given two navigation steps S_1 and S_2 :

1. **Decryption:** The cipher string is decoded into a Base-10 integer $N = \sum_{i=0}^{L-1} \phi^{-1}(c_i) \cdot 7^{(L-1-i)}$.
2. **Signed Interpretation:** FWD implies $+N$, while BACK implies $-N$.
3. **Arithmetic Combination:** We compute $R = N_1 + N_2$.
4. **Re-encryption:** The result R is encoded back into the target format:
 - If $R > 0$, the direction is FWD.
 - If $R < 0$, the direction is BACK.
 - The absolute value $|R|$ is converted to Base-7 and mapped via ϕ .

Example Computing $\text{FWD}(\text{ad}) + \text{BACK}(\text{ef})$:

- $\text{ad}_7 \rightarrow 0 \cdot 7 + 3 = 3$. Direction FWD $\rightarrow +3$.
- $\text{ef}_7 \rightarrow 4 \cdot 7 + 5 = 33$. Direction BACK $\rightarrow -33$.
- Sum: $3 + (-33) = -30$.
- Result: Magnitude $30 = 42_7 \rightarrow \text{ec}$. Sign Negative $\rightarrow \text{BACK}$. Final: $\text{BACK}(\text{ec})$.

C.2 Enigma 3-Rotor System (Finite Abelian Group)

This domain simulates the state of a simplified Enigma machine with three independent rotors. Mathematically, it represents the direct product group $\mathbb{Z}_{26} \times \mathbb{Z}_{26} \times \mathbb{Z}_{26}$.

Representation

- **State:** A comma-separated string R_1, R_2, R_3 where each $R_i \in \{A, \dots, Z\}$.
- **Mapping:** Letters map to offsets $0 \dots 25$ ($A=0, B=1, \dots, Z=25$).

Operational Logic The binary operation combines two states via component-wise addition modulo 26. Unlike a physical Enigma machine, there is no “carry” or stepping mechanism between rotors in this simplified algebraic model.

$$(r_1, r_2, r_3) \oplus (s_1, s_2, s_3) = ((r_1 + s_1) \bmod_{26}, (r_2 + s_2) \bmod_{26}, (r_3 + s_3) \bmod_{26}) \quad (10)$$

The inverse operation corresponds to negation modulo 26: $\text{inv}(r) = (26 - r) \pmod{26}$.

Example Computing $A, C, Z \oplus B, B, C$:

- Rotor 1: $A(0) + B(1) = 1 \rightarrow B$
- Rotor 2: $C(2) + B(1) = 3 \rightarrow D$
- Rotor 3: $Z(25) + C(2) = 27 \equiv 1 \pmod{26} \rightarrow B$
- Result: B, D, B

C.3 Knitting Instruction System (Infinite Non-Abelian Group)

This domain models a Free Group generated by two elements. It represents sequences of reversible actions where order matters and adjacent inverses cancel.

Representation

- **Alphabet:** $\Sigma = \{k, p, K, P\}$.
- **Semantics:** k (knit), p (purl). Capital letters represent inverses: $K = k^{-1}$ (un-knit), $P = p^{-1}$ (un-purl).
- **Identity:** The empty string ϵ .

Operational Logic The operation is string concatenation followed by recursive reduction. The reduction rule is defined by the cancellation of adjacent inverse pairs:

$$xx^{-1} \rightarrow \epsilon \quad \text{and} \quad x^{-1}x \rightarrow \epsilon \quad (11)$$

Specifically, the pairs (k, K) and (p, P) annihilate each other. This process repeats until the string is in its irreducible canonical form.

Example Computing $kp \oplus PK$:

1. Concatenate: $kpPK$
2. Inner Cancellation: The adjacent pair p, P cancels. $k(pP)K \rightarrow kK$.
3. Secondary Cancellation: The adjacent pair k, K cancels. $kK \rightarrow \epsilon$.
4. Result: Empty string (Identity).

C.4 Rubik's Cube Sequence System (Finite Non-Abelian Group)

This domain models the algebra of face rotations on a $3 \times 3 \times 3$ cube. It enforces strict canonicalization rules involving commutation and modulo arithmetic, representing a quotient of a free group.

Representation

- **Moves:** $F \in \{R, L, U, D, F, B\}$ representing face turns.
- **Modifiers:** None (90° clockwise), # (90° counter-clockwise/prime), 2 (180°).
- **Sequence:** A space-separated list of moves (e.g., $R \ U \ R\#$).

Operational Logic Operations are defined by sequence concatenation followed by reduction to a unique **Canonical Form**. Three specific rules are applied iteratively:

1. **Inverse Cancellation:** Moves followed by their inverse are removed (e.g., $RR\# \rightarrow \epsilon$).
2. **Face Consolidation (Mod 4):** Consecutive moves of the same face are summed modulo 4.
 - Example: $R \cdot R2 \rightarrow R3 \equiv R\#$ (since $1 + 2 = 3 \equiv -1$).
 - Example: $R \cdot R \cdot R \cdot R \rightarrow R4 \equiv \epsilon$.
3. **Commutative Reordering:** While the group is generally non-abelian, opposite faces commute (e.g., $RL = LR$). To ensure a unique representation, we enforce a lexicographical priority:
 - R precedes L ; U precedes D ; F precedes B .
 - Example: A sequence input as $L \ R$ is automatically rewritten to $R \ L$.

Example Solving equation $X = \text{inv}(R \ U) \cdot R \ U \ R\#$:

1. Invert $R \ U \rightarrow U\# \ R\#$.
2. Concatenate: $U\# \ R\# \ R \ U \ R\#$.
3. Consolidate: $R\#$ and R cancel. Sequence becomes $U\# \ U \ R\#$.
4. Consolidate: $U\#$ and U cancel.
5. Result: $R\#$.