WHEN CAN LLMS LEARN TO REASON WITH WEAK SUPERVISION?

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models have achieved significant reasoning improvements through reinforcement learning with verifiable rewards (RLVR). Yet as model capabilities grow, constructing high-quality reward signals become increasingly difficult, making it essential to understand when RLVR can succeed under weaker forms of supervision. We conduct a systematic empirical study across diverse model families and reasoning domains under three weak supervision settings: scarce data, noisy rewards, and self-supervised proxy rewards. We find that generalization is closely associated with training reward saturation dynamics: models that generalize exhibit a prolonged pre-saturation phase during which training reward and downstream performance climb together, while models that saturate rapidly memorize rather than learn. We identify reasoning faithfulness, defined as the extent to which a model's intermediate steps logically support its final answer, as the pre-RL property that predicts which regime a model falls into, while output diversity alone is uninformative. Motivated by these findings, we disentangle the contributions of continual pre-training and supervised fine-tuning, finding that SFT on explicit reasoning traces is necessary for generalization under weak supervision, while continual pre-training on domain data amplifies the effect. Applied together to Llama3.2-3B-Base, these interventions enable generalization across all three settings where the base model previously failed.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has emerged as a powerful paradigm for improving reasoning capabilities in large language models (Guo et al., 2025; Jaech et al., 2024; Team et al., 2025). With only binary feedback on correctness, RLVR has enabled substantial gains across diverse reasoning tasks without requiring dense supervision. However, recent findings suggest these improvements may be driven by factors other than the integration of correctness signals. Some studies report that RLVR succeeds even under extreme conditions: training on just a single example can yield significant gains (Wang et al., 2025a), and random or incorrect rewards sometimes match ground-truth performance (Shao et al., 2025). Other work shows that proxy signals such as self-certainty (Zhao et al., 2025; Prabhudesai et al., 2025), entropy minimization (Agarwal et al., 2025), majority voting (Zuo et al., 2025), or self-generated training data (Huang et al., 2025) can replace verifiable rewards.

Furthermore, techniques that succeed on one model family often fail on others (Shao et al., 2025), underreported baselines may inflate perceived benefits (Chandak et al., 2025), and prolonged training with proxy rewards (i.e., reward signals derived from model outputs without ground-truth verification) can lead to reward hacking and performance collapse (Shafayat et al., 2025). These mixed results leave a fundamental question: *When can RLVR generalize*[1] *under weak supervision, and what determines success or failure?*

Understanding when RLVR works under weak supervision matters in practice. Ground-truth verifiers are often limited: labels may be noisy or unavailable, and as models become stronger than their supervisors, alternative reward signals become necessary (Burns et al., 2023).

[1]Throughout, we use generalization to mean improvement on downstream evaluation benchmarks, both in-domain held-out sets and out-of-domain transfer, following RL training.

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

We conduct a systematic empirical study of RLVR under weak supervision across two model families (Qwen and Llama), and three reasoning domains (MATH, SCIENCE, and GRAPH). Our work is organized around three questions:

- **RQ1 (Weak Supervision):** When does RLVR generalize across model families and domains under scarce data, noisy rewards, and self-supervised proxy rewards?
- **RQ2 (Model Properties):** What pre-RL model properties determine whether a model generalizes under weak supervision?
- **RQ3 (Intervention):** How can we enable generalization in models that fail under weak supervision?

Our investigation uncovers three findings. **First, generalization under weak supervision is governed by training reward saturation dynamics**. Models that generalize exhibit a prolonged *pre-saturation* phase during which training reward climbs steadily and the model learns transferable reasoning patterns; models that fail saturate rapidly and enter a *post-saturation* phase where further training yields diminishing returns. Which regime a model falls into depends on its pretraining priors: models with strong domain-aligned pretraining (Qwen on MATH and SCIENCE) sustain extended pre-saturation phases and generalize under scarce data, noisy rewards, and self-supervised proxy rewards, while models without such priors (Llama across all domains, and Qwen on GRAPH) saturate rapidly and fail to generalize even under moderate label noise. We treat the model-family contrast as a proxy for pretraining-prior strength rather than an intrinsic property of either family, a reading that §4 confirms by showing that continual pre-training on math data transforms Llama's RL behavior to resemble Qwen's.

Second, reasoning faithfulness, not just output diversity, distinguishes models that generalize from models that memorize. A natural hypothesis for rapid saturation is that failing models lack exploratory capacity. We find the opposite: Llama models reach perfect training reward faster than Qwen *and* maintain higher output diversity throughout training, yet they generalize poorly. The missing property is reasoning faithfulness, defined by whether a model's intermediate steps support its final answer. Models that saturate rapidly produce diverse correct answers through reasoning chains that do not justify them. **Third, SFT on explicit reasoning traces is necessary for generalization under weak supervision, and continual pre-training amplifies the effect**. We run a controlled comparison that disentangles the two interventions, training Llama3.2-3B Base, a continually pre-trained variant (CPT, ours), and Instruct, each with either Thinking SFT (explicit reasoning traces) or Non-Thinking SFT (final solutions only). Thinking SFT is necessary: it improves reasoning faithfulness, extends the pre-saturation phase, and enables generalization under all three weak supervision settings, while Non-Thinking SFT on the same prompts fails. Continual pre-training is a multiplier rather than a substitute. CPT combined with Thinking SFT produces the strongest generalization, recovering performance in settings where Llama previously failed.

2 EXPERIMENTAL SETUP

We evaluate the following model families: (1) **Qwen2.5-1.5B / 3B** (Base): General-purpose models pretrained on 18 trillion tokens (Team, 2024); (2) **Qwen2.5-Math-1.5B / 7B** (Math-specialized): Built upon Qwen2.5 with an additional 1 trillion math-related tokens (Yang et al., 2024); (3) **Llama-3.2-3B / 8B-Instruct** (Instruction-tuned): Pretrained on 9 trillion tokens and aligned via SFT, rejection sampling, and DPO (Dubey et al., 2024). We use Instruct variants for Llama as the base models don't reliably follow the required format for on-policy rollouts. We revisit Llama-Base in §4 , where SFT handles the format-following issue. **Domains and Datasets.** We select three domains with varying levels of pretraining exposure: MATH (high exposure), SCIENCE (moderate coverage) and GRAPH tasks (underrepresented in typical pretraining corpora). We use Skywork-OR1 (He et al., 2025a) for MATH, SCP datasets (Liu et al., 2025a; Lu et al., 2025) spanning physics, chemistry, and biology for SCIENCE, and tasks from Reasoning Gym (Stojanovski et al., 2025) involving discrete algorithmic reasoning for GRAPH. For Math and Science, we use the 1.5B/3B models as our primary experiments and additionally evaluate 7B/8B models to verify that our findings hold at larger scale. For GRAPH, we only use the 7B/8B variants because the smaller models achieve solve@16 = 0, leaving no informative signal for RL. More details are provided in Appendix C.

Model-Aware Data Filtering. To ensure informative training signal, we implement model-specific difficulty filtering. For each problem, we sample 16 responses and count correct solutions (solve@16 $\in [0, 16]$). We retain only problems where solve@16 $\in [1, 15]$, discarding those either

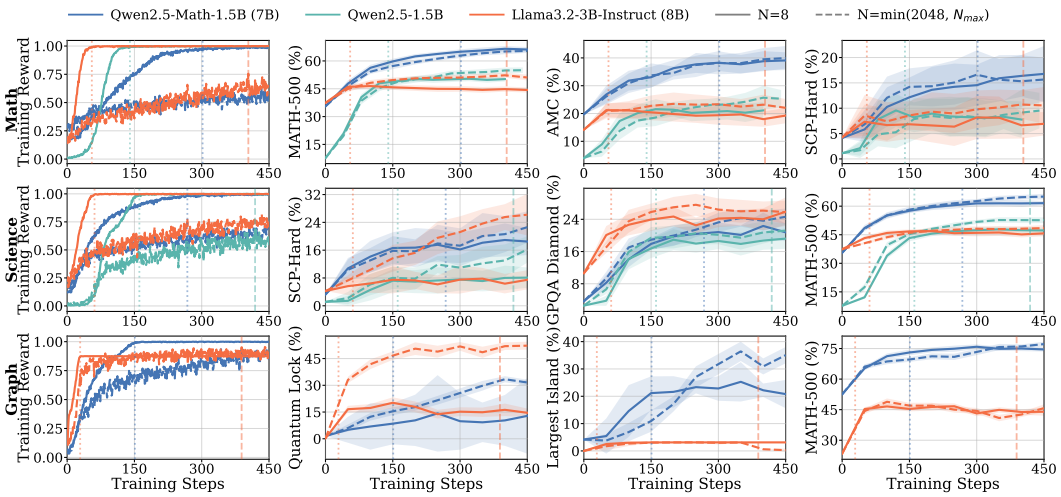


Figure 1: **Training dynamics and test performance (avg@16) across model families and domains.** Each row corresponds to one domain. Columns show training reward (col. 1), in-domain benchmarks (col. 2–3), and OOD benchmark (col. 4) over RL steps for two dataset sizes: $N = 8$ (solid lines) and $N = N_{\max}$ (dashed lines), where $N_{\max}$ is the largest available training set for the model in that domain ($N_{\max} = 2048$ for MATH/SCIENCE; 882 for Qwen and 256 for Llama on GRAPH). Vertical dashed lines mark the saturation step t^N_{sat} for each run; shaded regions indicate ± 1 standard deviation over independent sampling. **Qwen models exhibit extended pre-saturation phases and generalize from 8 samples, while Llama models saturate rapidly with limited gains.** 7B/8B model results in Appendix D.3.

trivial or intractable for the model, stratified equally across difficulty levels (details in Appendix C.2). This filtered set serves as the candidate pool for all weak supervision settings; we describe how training data is constructed from this pool for all settings in §3.

Training Configuration. We use GRPO (Group Relative Policy Optimization) as our RL algorithm (Shao et al., 2024). For each query q sampled from training datasets $\mathcal{D}$, a group of individual responses $\{o_i\}_{i=1}^G$ are sampled from the policy $\pi_{\theta_{\text{old}}}$ before the update. GRPO maximizes the objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \Bigg[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min\left(\rho_{i,t}\hat{A}_i, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_i\right) - \beta D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}) \Bigg]$$

where $\rho_{i,t} := \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ is the probability ratio between the current and old policy, and $\hat{A}_i := \frac{r_i - \text{mean}(\{r_i\}_{i=1}^G)}{\text{std}(\{r_i\}_{i=1}^G))}$ is the advantage of i-th response calculated by normalizing the group-level rewards. Rewards $r_i \in \{0, 1\}$ are binary and assigned by ground-truth answer verification. The KL regularization $D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}})$ is applied to a fixed reference policy π_{ref}, weighted by a scalar coefficient β. All experiments use the `verl` framework (Sheng et al., 2024) (hyperparameter details in Appendix C.3).

Evaluation. We evaluate reasoning performance using avg@16 accuracy (average pass@1 over 16 independent samples per problem) with temperature 1.0 sampling and report pass@k for $k \in \{4, 8, 16\}$ in the Appendix. For MATH, we use MATH-500, AMC, AIME 2024, AIME 2025, Minerva Math, and OlympiadBench evals. For SCIENCE, we use GPQA-Diamond, a held-out SCP-Hard set (Liu et al., 2025a) (a subset of SCP problems where both Qwen2.5-1.5B and Llama-3.2-3B-Instruct achieve solve@16 $= 1$ pre-RL), Science Bench, MMLU-Science, and SuperGPQA. For GRAPH, we use held-out Quantum Lock and Largest Island tasks from Reasoning Gym (Stojanovski et al., 2025), filtered similarly to solve@16 $= 1$. For each domain, we designate benchmarks as in-domain or out-of-domain (OOD); e.g., for MATH training, MATH-500 and AMC are in-domain while SCP-Hard and GPQA-Diamond are OOD (full assignments in Appendix Table 2). We report representative results in the main text and full results in the Appendix.

Table 1: **Comparison of saturation steps $t_{\text{sat}}^{(8)}$, pre-saturation gain $\Delta_{\text{sat}}^{(8)}$ and post-saturation residual $\Delta_{\text{post}}^{*(8)}$ across model families and training domains when training on 8 examples**. We additionally report the large-small gap $G_{\text{sat,in}}^{(n_1,8)}$ and $G_{\text{sat,ood}}^{(n_1,8)}$. For Graph, the largest available setting is $n_1 = 882$ for Qwen model and $n_1 = 256$ for Llama model (marked with †). The green cells mark $\Delta_{\text{sat}}^{(8)} > 0$ (effective pre-saturation learning) while red mark rapid saturation $t_{\text{sat}}^{(8)} < 100$. The large-small gap at saturation steps $G_{\text{sat,in}}^{(n_1,8)}$ and $G_{\text{sat,ood}}^{(n_1,8)}$ are generally small. Results on more benchmarks and pass@k metrics are reported in Table 3-7 in Appendix.

Model	$t_{\text{sat}}^{(8)}$	**In-domain Benchmarks**					**OOD Benchmark**		
		$\Delta_{\text{sat}}^{(8)}$	$\Delta_{\text{post}}^{*(8)}$	$\Delta_{\text{sat}}^{(8)}$	$\Delta_{\text{post}}^{*(8)}$	$G_{\text{sat,in}}$	$\Delta_{\text{sat}}^{(8)}$	$\Delta_{\text{post}}^{*(8)}$	$G_{\text{sat,ood}}$
Training Domain: Math		**MATH500**		**AMC**		$G_{\text{sat,in}}^{(2048,8)}$	**SCP-Hard**		$G_{\text{sat,ood}}^{(2048,8)}$
Qwen2.5-Math-1.5B	302	29.7	1.5	18.7	0.6	-1.1	10.5	2.1	2.4
Qwen2.5-1.5B	170	32.1	0.9	12.7	3.3	-0.5	7.0	0.3	-0.4
Llama3.2-3B-Instruct	55	10.8	-1.9	8.8	-2.1	-0.9	3.9	0.0	1.5
Training Domain: Science		**SCP-Hard**		**GPQA-Diamond**		$G_{\text{sat,in}}^{(2048,8)}$	**MATH500**		$G_{\text{sat,ood}}^{(2048,8)}$
Qwen2.5-Math-1.5B	268	14.5	1.1	16.9	1.6	1.1	25.3	0.8	1.1
Qwen2.5-1.5B	161	6.4	0.2	13.3	1.7	1.8	32.3	2.1	1.2
Llama3.2-3B-Instruct	61	1.8	1.7	11.9	3.0	5.1	7.3	2.2	0.6
Training Domain: Graph		**Quantum Lock**		**Largest Island**		$G_{\text{sat,in}}^{(n_1,8)\dagger}$	**MATH500**		$G_{\text{sat,ood}}^{(n_1,8)\dagger}$
Qwen2.5-Math-7B	150	8.3	4.9	19.8	1.9	-1.8	21.0	2.1	-3.7
Llama3.1-8B-Instruct	29	10.1	7.1	1.8	1.0	$3.0^{\dagger}$	9.1	3.8	$0.0^{\dagger}$

3 RLVR Under Weak Supervision

To understand when RLVR generalizes under weak supervision, we study three settings: **scarce data** (§3.1), **noisy rewards** (§3.2), and **self-supervised proxy rewards** (§3.3), then analyze **policy behavior** to explain why some models succeed and others fail (§3.4). Throughout, we compare Qwen and Llama as a proxy for variation in pretraining priors rather than an intrinsic property of either family: Qwen2.5-Math has an additional 1T math-specific pretraining tokens, while Llama-3.2-Instruct is aligned for general instruction-following. §4 confirms this reading by showing that continual pre-training on math data transforms Llama's RL behavior to resemble Qwen's. We additionally analyze GRPO baseline selection in Appendix F.

3.1 Scarce Data

To understand how data scarcity affects RLVR generalization, we investigate training dynamics across dataset sizes $N \in \{8, 32, 64, 512, 2048\}$. Unlike prior work on sample-efficient RLVR (Wang et al., 2025a; Sun et al., 2025), which select specific data points, we use stratified random sampling across difficulty levels defined in §2. For $N < 64$, we repeat prompts uniformly to reach batch size 64 (e.g., $N = 8$ implies 8 repeats).

To study training dynamics, we leverage reward saturation to distinguish periods where the policy improves on the training dataset from those where it plateaus. Intuitively, once training reward saturates, further updates yield little new signal. We define $\bar{r}_t := \mathbb{E}_{q\sim\mathcal{D},\,\{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G} r_i\right]$ as the expected training reward at update step $t \in \{1, \ldots, T\}$, and let $\bar{r}_{\max} := \max_{1\le t\le T} \bar{r}_t$ be the maximum reward observed during training. We define the *saturation step* as the earliest update at which the reward is close to this maximum: $t_{\text{sat}} := \min\left\{t \in \{1, \ldots, T_{\text{eff}}\} : \bar{r}_t \ge \epsilon_{\max}\bar{r}_{\max}\right\}$.

We use $\epsilon_{\max} = 0.99$ and set $T_{\text{eff}} = T - 50$, i.e., we search for t_{sat} only up to the first T_{eff} updates to avoid boundary effects near the end of training. We define the ***pre-saturation phase*** as all steps $t \in \{1, \ldots, t_{\text{sat}} - 1\}$ and ***post-saturation phase*** as all steps $t \in \{\min(t_{\text{sat}}, T), \ldots, T\}$.

To quantify data efficiency, we introduce three metrics. Let $M^{(n)}(t)$ denote an evaluation metric (e.g., avg@16 on MATH-500) at training step t with n samples, and $t_{\text{sat}}^{(n)}$ the corresponding saturation step.

- **Pre-saturation gain** $\Delta_{\text{sat}}^{(n)}(M)$: performance gain from initialization to saturation as $\Delta_{\text{sat}}^{(n)}(M) := M^{(n)}\big(t_{\text{sat}}^{(n)}\big) - M^{(n)}(0)$. Larger positive values indicate effective learning before saturation.
- **Post-saturation residual**: $\Delta_{\text{post}}^{*(n)}(M) := \max_{t \in [t_{\text{sat}}^{(n)}, T]} M^{(n)}(t) - M^{(n)}\big(t_{\text{sat}}^{(n)}\big)$, the maximum additional gain after saturation. Values near zero indicate negligible post-saturation gains.
- **Large-small gap**: $G_{\text{sat}}^{(n',n)}(M) := M^{(n')}(t_{\text{sat}}^{(n)}) - M^{(n)}(t_{\text{sat}}^{(n)})$ for $n' > n$, which compares the larger (n') and smaller (n) dataset runs at the smaller run's saturation step. Values near zero suggest limited advantage from increasing dataset size. We denote $G_{\text{sat,in}}^{(n_1,8)}$ and $G_{\text{sat,ood}}^{(n_1,8)}$ as the average gaps over in-domain and OOD benchmarks, respectively.

Pre-saturation phase dominates small-sample learning, and its length predicts generalization. Table 1 summarizes the proposed metrics across model families and training domains when training on 8 examples (additional benchmarks and pass@k metrics in Appendix D.2, Tables 3–7). All model-domain pairs show clearly positive $\Delta_{\text{sat}}^{(8)}$ (both avg@16 and pass@k, $k \in \{4, 8, 16\}$) across in-domain and out-of-domain benchmarks, indicating that as few as 8 training examples can trigger measurable learning during the pre-saturation phase. The large-small gap $G_{\text{sat}}^{(2048,8)}$ is near zero for 7 of 8 model-domain pairs, suggesting that early learning is not strongly data-limited. In contrast, $\Delta_{\text{post}}^{*(8)}$ is typically smaller than $\Delta_{\text{sat}}^{(8)}$, indicating diminishing returns after saturation.

Fig. 1 shows the training curves across data scales. With 8 training samples, Qwen2.5-Math-1.5B on MATH increases reward steadily for over 300 steps, allowing the model to extract generalizable reasoning patterns that transfer to held-out benchmarks such as MATH-500 and SCP-Hard. A within-family comparison isolates the pretraining effect: Qwen2.5-Math-1.5B, which shares architecture with Qwen2.5-1.5B but has additional math-specific pretraining, saturates more slowly and transfers further (Table 1). Full data-scale results ($N \in \{8, 32, 64, 512, 2048\}$) in Appendix D.1 confirm that Qwen's performance on MATH and SCIENCE is nearly independent of N, while Llama across all domains and Qwen on GRAPH show clear data dependence.

Models without domain-aligned priors saturate rapidly and fail to generalize. In contrast, Llama models across all domains and Qwen on GRAPH (Fig. 1) exhibit clear dependence on data scale. For Llama, training on 8 samples leads to rapid saturation ($t_{\text{sat}}^{(8)} < 100$), and meaningful generalization requires $N \geq 512$ (Appendix D.1: Figs. 9 and 10). Even Qwen2.5-Math with strong math priors saturates rapidly on GRAPH, suggesting that domain-specific pretraining is necessary to sustain the pre-saturation phase. 7B/8B model results confirming these trends are in Appendix D.3.

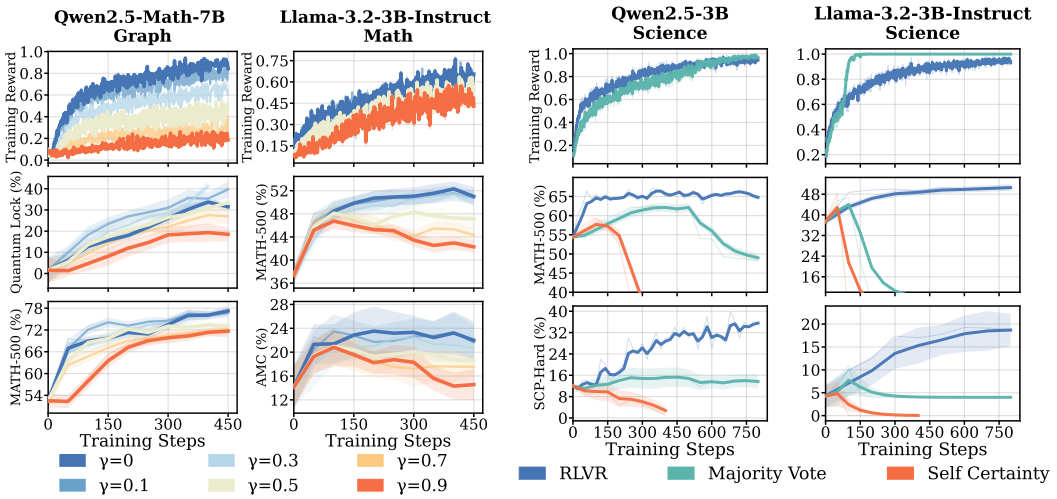
Extended pre-saturation enables out-of-domain transfer. Positive $\Delta_{\text{sat}}^{(8)}$ values in Table 1 indicate that reasoning patterns learned during the pre-saturation phase transfer across domains, particularly for Qwen models. With only 8 samples, Qwen2.5-1.5B trained on MATH achieves consistent gains on the out-of-domain SCIENCE benchmark (SCP-Hard), while Qwen2.5-Math-7B trained on GRAPH improves out-of-domain MATH-500 performance by 21.0% (Fig. 1). In contrast, Llama models show limited out-of-domain transfer even when in-domain performance improves.

Takeaway: (1) RLVR can generalize from as few as 8 samples when models remain in an extended pre-saturation phase, whereas rapidly saturating models require substantially more data. (2) Whether scarce-data learning succeeds is model- and domain-dependent, reflecting pretraining priors. (3) In the low-data regime, Llama models saturate training rewards much faster than Qwen by memorizing training examples but achieve little meaningful task learning.

3.2 NOISY REWARDS

When ground-truth verifiers are available but imperfect, reward labels may contain errors. To evaluate RLVR robustness to such noisy supervision, we vary the fraction of incorrect labels γ by randomly replacing ground-truth answers with the most frequent incorrect answer produced by the model itself (details in Appendix E.1). Unless otherwise noted, experiments use $N = 2048$.

RLVR demonstrates robustness to reward noise, but generalization varies across models. Fig. 2(a) and Appendix Fig. 22 summarize performance across seven model–domain pairs under varying γ. At $\gamma \leq 0.3$, test performance remains close to clean rewards ($\gamma = 0$), indicating robustness to moderate noise. On MATH and SCIENCE, Qwen models maintain gains under substantial corruption



(a) Noisy rewards

(b) Self-supervised proxy rewards

Figure 2: **(a) Noisy rewards:** Effect of reward label corruption on training dynamics. γ denotes the fraction of corrupted labels, from clean ($\gamma = 0$) to mostly incorrect ($\gamma = 0.9$). For Qwen on GRAPH and Llama on MATH, generalization degrades at $\gamma \geq 0.5$. For Llama, training reward curves stay close across all γ, suggesting overfitting to noise. **(b) Self-supervised proxy rewards:** Comparison of RLVR, self-certainty, and majority vote with 1024 training samples. Proxy rewards without verifiers exhibit failure modes under prolonged training: collapse (self-certainty) and reward spikes followed by performance drops (majority vote). More results in Appendix E.2.

(up to $\gamma = 0.7$), while Qwen on GRAPH and Llama on MATH and SCIENCE degrade at $\gamma \geq 0.5$. Notably, Llama on MATH shows nearly identical reward curves across all γ, indicating it fits incorrect answers as easily as correct ones. Model-domain pairs with faster saturation (§3.1) are generally less robust to label noise, a connection we develop in §3.4 and §4.

Takeaway: Robustness to label noise varies sharply: Qwen on MATH and SCIENCE tolerates up to 70% corruption, while Llama and Qwen on GRAPH degrade at 50%. Faster-saturating model-domain pairs are less robust, and Llama fits corrupted labels nearly as fast as clean ones, suggesting that rapid saturation reflects memorization rather than learning.

3.3 SELF-SUPERVISED PROXY REWARDS

When ground-truth verifiers are entirely unavailable, models must rely on alternative reward signals (Burns et al., 2023; Rahman et al., 2025; Bowman et al., 2022). We evaluate two self-supervised proxy rewards derived from model outputs: self-certainty (Zhao et al., 2025) and majority vote (Zuo et al., 2025) (details in Appendix E.2). **Proxy rewards trigger reward hacking and policy collapse.** Fig. 2(b) shows that fully replacing verifiable feedback with self-supervised proxy signals introduces severe failures under prolonged training. Only math-specialized models (Qwen2.5-Math-1.5B on MATH and SCIENCE) show improvement with majority voting, while other models fail entirely. For Qwen2.5-3B on SCIENCE, majority voting yields temporary gains before collapse after $\sim$500 steps, as the policy converges toward a single output to maximize agreement (see Table 8 for an example). Self-certainty leads to performance collapse across all settings (details in Appendix E.2, Fig. 23).

Takeaway: Self-supervised proxy rewards succeed only for math-specialized models (Qwen-Math under majority voting). Other models exhibit the same pattern as in §3.1 and §3.2: faster saturation and weaker pretraining priors coincide with brittleness, and under prolonged training, policies converge toward outputs that maximize the proxy without downstream gains.

3.4 WHY DO MODELS FAIL UNDER WEAK SUPERVISION?

The results in §3.1–§3.3 show a consistent pattern: models with strong domain-aligned pretraining (Qwen on MATH and SCIENCE) generalize under weak supervision, while those without (Llama

across domains, Qwen on GRAPH) fail. A natural hypothesis, motivated by prior work linking diminished exploration to rapid saturation (Cui et al., 2025), is that failing models produce less diverse outputs. To test this, we analyze model behavior along two axes: *response diversity* and *reasoning faithfulness*.

High diversity does not prevent rapid saturation. We measure semantic diversity (Farquhar et al., 2024; Li et al., 2025) by clustering model responses via pairwise LLM-judge similarity and computing the Shannon diversity index (full methodology and figures in Appendix G.1). Llama reaches reward saturation earlier yet retains higher diversity than Qwen throughout training (Fig. 26(a)), the opposite of what the exploration-saturation hypothesis predicts.

Since diversity alone does not explain failure, we investigate *reasoning faithfulness* (Baker et al., 2025): whether a model's reasoning trace contains the information needed to justify its final answer and is logically consistent with it. We categorize each rollout as *aligned*, *partially aligned*, or *misaligned* via an LLM-as-a-judge and compute the faithfulness rate $F_\pi(l)$ as the fraction of responses assigned to label l (methodology and inter-judge agreement in Appendix G.2).

Models with rapid saturation exhibit low reasoning faithfulness. Fig. 26(b) shows the fraction of correct responses that are *aligned* over RL training with 8 samples. On MATH, Llama shows much lower faithfulness than Qwen, indicating that its rapid reward gains reflect memorization rather than improved reasoning (additional results in Appendix G, Fig. 32).

Reasoning diversity should be considered jointly with faithfulness. Fig. 26(c) reports *faithful diversity*: diversity computed only over faithful responses. On MATH, Llama's apparent diversity advantage disappears as most diverse responses are unfaithful, and the faithful subset is narrow. On SCIENCE and GRAPH, faithful diversity separates models that raw alignment proportions cannot distinguish (Fig. 29). In every case, the model that generalizes best is the one exploring the widest range of faithful reasoning paths.

Takeaway: Low reasoning faithfulness, not low diversity, explains failure under weak supervision: rapidly saturating models memorize answers rather than learn transferable reasoning. Raw diversity metrics mislead, as Llama exhibits higher output diversity than Qwen while generalizing worse. Diversity becomes informative only when computed over faithful responses.

4 IMPROVING RLVR UNDER WEAK SUPERVISION VIA PRE-RL TRAINING

Section 3 showed that rapid saturation and low reasoning faithfulness are linked: models that generalize poorly under weak supervision produce correct answers through reasoning that does not support them. This raises a natural question. If faithfulness is associated with a longer pre-saturation phase, and furthermore with generalization, then pre-RL interventions that raise faithfulness should also extend the phase and improve generalization. We test this by running a controlled comparison of pre-RL interventions on Llama3.2-3B, the model that failed most consistently in §3.

We study two axes of pre-RL training. The first is continual pre-training (CPT), extended training on domain-specific pretraining tokens to strengthen the pretraining prior. The second is supervised fine-tuning (SFT), with the specific question of whether SFT on explicit reasoning traces differs in its effect from SFT on final answers alone. Crossing these axes gives a 2×2 design: two initializations (Base, CPT) each followed by two SFT regimes (Thinking, Non-Thinking). We additionally include Llama3.2-3B-Instruct as a reference: it shares the architecture of Llama3.2-3B-Base but has undergone extensive instruction tuning, rejection sampling, and DPO, providing a strong off-the-shelf baseline against which to judge our targeted interventions. We then run RL under all three weak supervision settings from §3: scarce data, noisy rewards, and self-supervised proxy rewards.

We focus on the MATH domain for two reasons: Llama's baseline failure is sharpest there, providing the cleanest test of whether pre-RL interventions can recover generalization; and high-quality math pretraining corpora (Nemotron-CC-Math) and reasoning-trace datasets (OpenThoughts-114K) are available, enabling the interventions at sufficient scale.

Continual Pre-Training (CPT). We continually pre-train Llama3.2-3B-Base for one epoch on ∼52B math tokens from the Nemotron-CC-Math dataset (Mahabadi et al., 2025) (training details in Appendix C.5).

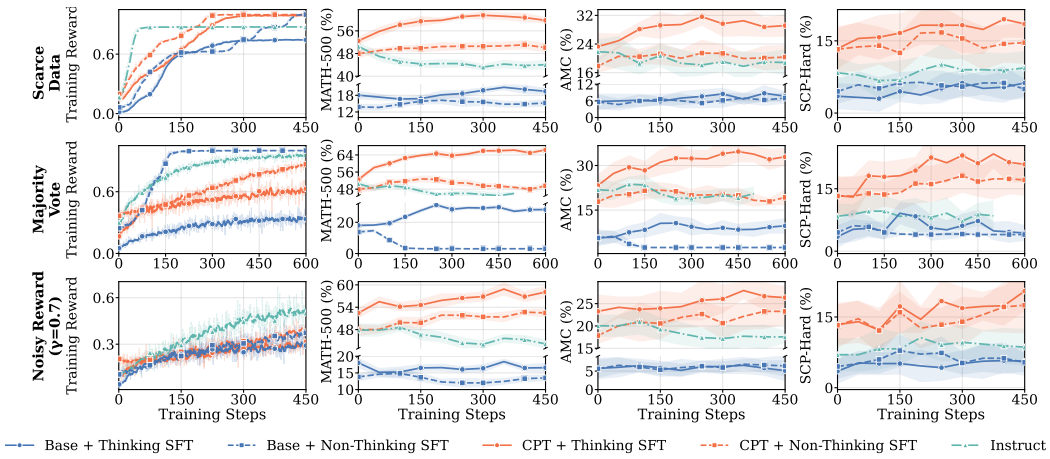


Figure 3: **RL training dynamics and generalization on MATH for Llama3.2-3B Base, CPT, and Instruct variants under different SFT initializations across three weak supervision settings: scarce data ($N = 8$, top), majority vote (middle), and noisy reward ($\gamma = 0.7$, bottom).** Thinking SFT (solid lines) consistently prolongs the pre-saturation phase and improves generalization for both CPT and Base models over Non-Thinking SFT counterparts (dashed lines) and the Instruct baseline (dash-dot). CPT + Thinking SFT achieves the strongest performance across all settings.

SFT Training Regimes. Following CPT or Base initialization, we compare two SFT regimes to determine whether explicit reasoning traces influence subsequent RL dynamics: **Non-Thinking SFT** supervises the model to output the final solution without reasoning traces, while **Thinking SFT** trains on explicit, verified long-form reasoning traces. Both use the same 43.5K math prompts from OpenThoughts-114K (Guha et al., 2025) (filtered to correct final answers and length below 8192 tokens). The SFT regimes are near-iso-compute: Thinking SFT trains on roughly 1B tokens, Non-Thinking SFT on roughly 0.27B, both negligible relative to the 52B-token CPT stage. Differences therefore reflect the content of the supervision rather than its cost (training example in Fig. 8; loss curves in Figs. 6 and 7). For the subsequent RL phase, we evaluate across all three weak supervision settings with a maximum response length of 8192 tokens; all other hyperparameters follow §2. Additional details in Appendix C.6.

4.1 RESULTS

Fig. 3 reports RL training dynamics across the three weak supervision settings, plotting training reward alongside two in-domain (MATH-500, AMC) and one out-of-domain (SCP-Hard) metrics (additional benchmarks and pass@k in Appendix H). We draw four findings:

Thinking SFT is necessary for substantial learning under weak supervision. The Instruct baseline is flat or decreasing across all three settings on all downstream evaluations. Thinking SFT is the only intervention that enables substantial downstream gains on scarce data and majority vote, and it does so for both Base and CPT initializations (solid blue and solid orange). Non-Thinking SFT shows modest gains only when paired with CPT, and only under noisy rewards; Non-Thinking SFT on Base is flat or degrades across all three settings.

CPT amplifies the Thinking SFT effect. Thinking SFT on Base alone produces modest gains; combined with CPT, it produces substantially larger final performance on every evaluation, making CPT + Thinking SFT the top-performing configuration across all settings. The CPT + Non-Thinking SFT comparison rules out a compute-based explanation: the same 52B CPT tokens, paired with SFT targets that strip reasoning traces, fail to enable generalization on scarce data and majority vote. The amplification is specific to the combination: extra pre-training compute alone is insufficient; Thinking SFT alone helps but is limited; only the combination recovers full generalization.

Base initialization fails under most weak supervision settings regardless of SFT. Base + Thinking SFT shows modest gains only under scarce data and majority vote; under noisy rewards, neither SFT

variant improves the Base model. This isolates CPT's contribution: Thinking SFT is necessary but not sufficient; domain-aligned pretraining is required to generalize across all three settings.

Thinking SFT improves reasoning faithfulness. In §3.4, we identified low reasoning faithfulness as the pre-RL property that distinguished failing from succeeding models. Fig. 28 shows that Thinking SFT raises aligned-response rate throughout the pre-saturation phase, relative to the Non-Thinking SFT baseline. CPT + Thinking SFT achieves the highest faithfulness among all configurations, consistent with its strongest generalization. Together with the extended pre-saturation dynamics in Fig. 3, this supports our hypothesis: pre-RL interventions that instill faithfulness produce longer pre-saturation phases and recover generalization in models that previously failed.

Takeaway: SFT on explicit reasoning traces, not final answers, is necessary for Llama to learn from RL under weak supervision. It recovers generalization under scarce data, noisy rewards, and self-supervised proxy rewards by raising reasoning faithfulness and extending the pre-saturation phase. Continual pre-training amplifies the effect but does not substitute for it: CPT + Non-Thinking SFT fails despite matched compute. The strongest configuration, CPT + Thinking SFT, recovers performance in settings where Llama previously collapsed entirely.

5 Related Work

RLVR has become an effective method for improving reasoning in LLMs (Guo et al., 2025; Yu et al., 2025; Zeng et al., 2025), with recent work showing that even a single training example (Wang et al., 2025a) or proxy rewards such as self-certainty (Zhao et al., 2025), majority voting (Zuo et al., 2025), and spurious signals (Shao et al., 2025) can yield gains. However, these findings often do not transfer across model families (Zeng et al., 2025; Shao et al., 2025). Concurrent work (He et al., 2026; Yang et al., 2026; Plesner et al., 2026) has studied RLVR under noisy or self-supervised rewards, but focuses on specific supervision settings or targeted fixes rather than explaining why weak supervision succeeds for some model-domain pairs and fails for others. Our work provides a unified explanation, showing that saturation dynamics governed by reasoning faithfulness determine generalization across model families and domains, and identifies a concrete pre-RL intervention that recovers generalization where it otherwise collapses.

Separately, recent work has studied how pre-training and mid-training shape RL outcomes (Qi et al., 2025; Wang et al., 2025b; Zhang et al., 2025; Akter et al., 2025), but focuses on compute allocation rather than enabling weak supervision. Research on reasoning faithfulness has documented mismatches between chain-of-thought traces and model predictions (Turpin et al., 2023; Chen et al., 2025b; Baker et al., 2025; Tutek et al., 2025; Gui et al., 2026), while work on RL diversity (Kirk et al., 2024; Yu et al., 2025) has linked exploration to generalization. Our work connects these lines: we show that diversity alone is uninformative, that faithfulness is the distinguishing property, and that pre-RL interventions targeting faithfulness extend the learning phase and recover generalization under weak supervision. Extended related work in Appendix B.

6 Conclusion

In this work, we studied when and why RLVR generalizes under weak supervision across diverse model families and three reasoning domains. Success under scarce data, noisy rewards, and self-supervised proxy rewards depends on pre-RL properties, pretraining priors and reasoning faithfulness, rather than on RL dynamics alone, and models that saturate rapidly produce correct answers through reasoning that does not support them. Pre-RL interventions targeting reasoning faithfulness recover generalization: SFT on explicit reasoning traces is the necessary ingredient, and continual pre-training on reasoning-heavy data amplifies the effect without substituting for it. These findings suggest two concrete practices for RL from weak supervision. First, monitor training reward saturation as a diagnostic: plateaued reward with flat downstream performance indicates the model has exhausted what RL can extract from its priors, and further RL compute is unlikely to help. Second, when weak supervision fails, allocate compute to pre-RL interventions that install strong priors rather than to longer RL training. Taken together, our findings argue that RL under weak supervision is best understood not as a training technique applied to a fixed model, but as the final stage of a pipeline whose success is largely determined before RL begins.

AI USE STATEMENT

We used generative AI tools to support writing and editing, primarily for correcting typographical and grammatical errors. We also used AI assistance for code implementation and debugging. All AI-assisted content was directed, reviewed, and verified by the authors, who take full responsibility for the final manuscript and code.

REFERENCES

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*, 2025.

AI-MO. Aime 2024. https://huggingface.co/datasets/AI-MO/aimo-validation-aime, 2024a.

AI-MO. Amc 2023. https://huggingface.co/datasets/AI-MO/aimo-validation-amc, 2024b.

Syeda Nahida Akter, Shrimai Prabhumoye, Eric Nyberg, Mostofa Patwary, Mohammad Shoeybi, Yejin Choi, and Bryan Catanzaro. Front-loading reasoning: The synergy between pretraining and post-training data. *arXiv preprint arXiv:2510.03264*, 2025.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*, 2025.

Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiūtė, Amanda Askell, Andy Jones, Anna Chen, et al. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Tong Wang, Samuel Marks, Charbel-Raphaël Ségerie, Micah Carroll, Andi Peng, Phillip J. K. Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca D. Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=bx24KpJ4Eb.

Nikhil Chandak, Shashwat Goel, and Ameya Prabhu. Incorrect baseline evaluations call into question recent llm-rl claims. https://safe-lip-9a8.notion.site/Incorrect-Baseline-Evaluations-Call-into-Question-Recent-LLM-RL-Claims-2012f1fbf0ee8094ab8ded1953c15a37?pvs=4, 2025. Notion Blog.

Peter Chen, Xiaopeng Li, Ziniu Li, Wotao Yin, Xi Chen, and Tianyi Lin. Exploration vs exploitation: Rethinking rlvr through clipping, entropy, and spurious reward. *arXiv preprint arXiv:2512.16912*, 2025a.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025b.

Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, Feng Yao, Yuexin Bian, Yonghao Zhuang, Nilabjo Dey, Yuheng Zha, Yi Gu, Kun Zhou, Yuqi Wang, Yuan Li, Richard Fan, Jianshu She, Chengqian Gao, Abulhair Saparov, Haonan Li, Taylor W. Killian, Mikhail Yurochkin, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. Revisiting reinforcement learning for llm reasoning from a cross-domain perspective, 2025. URL https://arxiv.org/abs/2506.14965.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

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

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

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. *arXiv preprint arXiv:2503.01307*, 2025.

Google DeepMind. Gemini 3 flash. `https://gemini.google.com/`, 2025. Released December 2025. Accessed: 2026-02-18.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

Runquan Gui, Yafu Li, Xiaoye Qu, Ziyan Liu, Yeqiu Cheng, and Yu Cheng. Faithrl: Learning to reason faithfully through step-level faithfulness maximization. *arXiv preprint arXiv:2602.03507*, 2026.

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

Bingxiang He, Yuxin Zuo, Zeyuan Liu, Shangziqi Zhao, Zixuan Fu, Junlin Yang, Cheng Qian, Kaiyan Zhang, Yuchen Fan, Ganqu Cui, et al. How far can unsupervised rlvr scale llm training? *arXiv preprint arXiv:2603.08660*, 2026.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025a.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Yang Liu, and Yahui Zhou. Skywork open reasoner series. `https://capricious-hydrogen-41c.notion.site/Skywork-Open-Reaonser-Series-1d0bc9ae823a80459b46c149e4f51680`, 2025b. Notion Blog.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Jian Hu, Mingjie Liu, Ximing Lu, Fang Wu, Zaid Harchaoui, Shizhe Diao, Yejin Choi, Pavlo Molchanov, Jun Yang, Jan Kautz, et al. Brorl: Scaling reinforcement learning via broadened exploration. *arXiv preprint arXiv:2510.01180*, 2025.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. *arXiv preprint arXiv:2508.05004*, 2025.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PXD3FAVHJT.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025a.

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. *arXiv preprint arXiv:2503.20783*, 2025b.

Dakuan Lu, Xiaoyu Tan, Rui Xu, Tianchu Yao, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. Scp-116k: A high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain. *arXiv preprint arXiv:2501.15587*, 2025.

Rabeeh Karimi Mahabadi, Sanjeev Satheesh, Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc-math: A 133 billion-token-scale high quality math pretraining dataset. *arXiv preprint arXiv:2508.15096*, 2025.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

OpenAI. Openai o3 and o4-mini system card. https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf, April 2025a. Accessed 2026-01-25.

OpenAI. gpt-oss-20b model card, 2025b. URL https://arxiv.org/abs/2508.10925. Accessed: 2026-02-18.

Opencompass. Aime 2025. https://huggingface.co/datasets/opencompass/AIME2025, 2024.

Andreas Plesner, Francisco Guzmán, and Anish Athalye. An imperfect verifier is good enough: Learning with noisy rewards. *arXiv preprint arXiv:2604.07666*, 2026.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

Zhenting Qi, Fan Nie, Alexandre Alahi, James Zou, Himabindu Lakkaraju, Yilun Du, Eric Xing, Sham Kakade, and Hanlin Zhang. Evolm: In search of lost language model training dynamics. *arXiv preprint arXiv:2506.16029*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023.

Salman Rahman, Sheriff Issaka, Ashima Suvarna, Genglin Liu, James Shiffer, Jaeyoung Lee, Md Rizwan Parvez, Hamid Palangi, Shi Feng, Nanyun Peng, et al. Ai debate aids assessment of controversial claims. *arXiv preprint arXiv:2506.02175*, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Sheikh Shafayat, Fahim Tajwar, Ruslan Salakhutdinov, Jeff Schneider, and Andrea Zanette. Can large reasoning models self-train? *arXiv preprint arXiv:2505.21444*, 2025.

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27 (3):379–423, 1948.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2505.24760*, 2025.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *arXiv preprint arXiv:2506.05316*, 2025.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasovic, and Yonatan Belinkov. Measuring chain of thought faithfulness by unlearning reasoning steps. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9935–9960, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.504. URL https://aclanthology.org/2025.emnlp-main.504/.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*, 2025b.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Shenzhi Yang, Guangcheng Zhu, Bowen Song, Sharon Li, Haobo Wang, Xing Zheng, Yingfan Ma, Zhongqi Chen, Weiqiang Wang, and Gang Chen. Can llms learn to reason robustly under noisy supervision? *arXiv preprint arXiv:2604.03993*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

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 preprint arXiv:2504.13837*, 2025.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and rl on reasoning language models. *arXiv preprint arXiv:2512.07783*, 2025.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.

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.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

A Limitations and Future Work

We acknowledge several limitations. First, due to computational constraints, our analysis is restricted to specific model families and scales. Validating these findings across larger architectures and broader task suites remains an important direction. Second, our analysis of diversity and faithfulness relies on an LLM-as-a-judge framework. Although we conducted small-scale human verification to validate label quality, we currently restrict this evaluation to a small scale to allow for reasonable labeling costs. Consequently, the development of scalable metrics for reasoning faithfulness and diversity remains an important direction for future research.

B Extended Related Work

RLVR for Reasoning. Reinforcement learning with verifiable rewards has emerged as an effective post-training method for improving reasoning in large language models (Guo et al., 2025; Olmo et al., 2025; Yu et al., 2025; Zeng et al., 2025). Recent work has explored when RLVR yields improvements (Liu et al., 2025b;a; Hu et al., 2025). Wang et al. (2025a) demonstrate that training on a single example can provide meaningful learning signals. Other work explores alternative rewards, including self-certainty (Zhao et al., 2025), majority voting (Zuo et al., 2025), negative signals (Zhu et al., 2025), self-generated training data Huang et al. (2025), and spurious rewards (Shao et al., 2025). However, these findings often do not transfer across model families, with studies reporting inconsistent results between Qwen and Llama (Zeng et al., 2025; Gandhi et al., 2025; Shao et al., 2025). Moreover, most prior work focuses on improving performance on narrow domains (primarily math) without examining generalization. Recent work (He et al., 2026; Yang et al., 2026; Plesner et al., 2026) has concurrently studied when and how RLVR can learn under self-supervision or noisy supervision. Our work extends this literature in two ways. First, we characterize the conditions under which RLVR generalizes across model families and domains, focusing on saturation dynamics and reasoning faithfulness. Second, we identify a concrete intervention that restores generalization in models where weak supervision would otherwise fail.

Role of Pre-Training and Fine-Tuning in RL. Recent work emphasizes that pre-training and mid-training shape RL generalization (Qi et al., 2025; Wang et al., 2025b; Zhang et al., 2025; Akter et al., 2025), but focuses on compute allocation and distribution alignment to improve performance. Our work specifically focuses on understanding how base model priors shaped from continual pretraining and reasoning SFT can enable generalization across different weak supervision settings.

Diversity and Faithfulness in Reasoning. Maintaining output diversity during RL has been proposed to promote exploration and mitigate model collapse (Kirk et al., 2024; Casper et al., 2023; Rafailov et al., 2023; Yu et al., 2025), but prior work has not explored what types of diversity benefit generalization. Separately, research has highlighted mismatches between chain-of-thought traces and model predictions (Turpin et al., 2023; Chen et al., 2025b; Baker et al., 2025; Tutek et al., 2025) and emphasized the importance of ensuring faithful reasoning throughout training (Gui et al., 2026). Wen et al. (2025) argues that RLVR can incentivize correct reasoning in base LLMs as long as priors have been established. Our work connects these lines of research, showing that diversity alone does not ensure generalization and that reasoning faithfulness distinguishes models' training dynamics. We further demonstrate that pre-RL intervention can improve reasoning faithfulness and improve generalization under weak supervision.

C Implementation Details

C.1 Training and Evaluation Datasets

We investigate RL training dynamics across two model families: Qwen (comprising Qwen2.5-1.5B/3B and Qwen2.5-Math-1.5B/7B) and Llama (Llama-3.2-3B/8B-Instruct). Our analysis spans three distinct reasoning domains, Math, Science, and Graph, allowing for a holistic investigation of RLVR under weak supervision across different domains and model families. For Math, we sample training prompts from the Skywork-OR1 (He et al., 2025b) dataset. For Science, we draw problems from the SCP dataset curated by prior work (Liu et al., 2025a; Lu et al., 2025), by selecting Physics, Chemistry, and Biology subjects. For Graph, we generate two synthetic

algorithmic tasks, *Quantum Lock* and *Largest Island*, using the curriculum specifications provided by the Reasoning Gym benchmark (Stojanovski et al., 2025). For each task, we instantiate five difficulty levels following the benchmark's curriculum, with a balanced number of samples per level.

We include the following domain-specific benchmarks for evaluations:

- **MATH500** (Lightman et al., 2023): A widely used subset of the MATH test split (Hendrycks et al., 2021).
- **AMC** (AI-MO, 2024b): 40 competition-level math questions.
- **AIME 2024** (AI-MO, 2024a): 30 competition-level math questions.
- **AIME 2025** (Opencompass, 2024): 30 competition-level math questions.
- **Minerva Math** (Lewkowycz et al., 2022): A set of 272 undergraduate-level science and math questions from MIT OpenCourseWare.
- **OlympiadBench** (He et al., 2024): A benchmark of 675 problems from international math olympiads and physics contests.
- **GPQA-Diamond** (Rein et al., 2024): 198 expert-level questions from GPQA spanning physics, chemistry, and biology; we preprocess the data following previous practice (Cheng et al., 2025).
- **SCP-Hard** (Lu et al., 2025; Liu et al., 2025a): A held-out set of 50 SCP questions filtered such that the base models (Qwen2.5-1.5B series models and Llama3.2-3B-Instruct model) achieve solve@16$=1$, containing disjoint questions from the SCP training datasets.
- **SuperGPQA** (Du et al., 2025): a subset constructed from the original SuperGPQA which contains 319 science questions and 250 non-science questions.
- **MMLU SCI** (Wang et al., 2024): a subset of MMLU Pro benchmark containing all college-level chemistry, physics and biology questions.
- **Science Bench** (Wang et al., 2023): 692 college-level science questions.
- **Graph Test**: A held-out set of 50 algorithmically generated instances from the *Quantum Lock* and *Largest Island* tasks using Reasoning Gym (Stojanovski et al., 2025), disjoint from training, filtered such that the base models (Qwen2.5-1.5B series and Llama3.2-3B-Instruct) achieve Pass@16$=1$.

  We also note that **GPQA-Diamond, MMLU SCI, and SuperGPQA are multiple-choice benchmarks, for which pass@k may be a less reliable metric**.

  Table 2 details the training and evaluation datasets across the three reasoning domains.

Table 2: **Training datasets and evaluation benchmarks across three reasoning domains.**

Domain	Training Source	In-Distribution Eval	Out-of-Distribution Eval
MATH	Skywork-OR1	MATH-500, AMC, AIME-2024, AIME-2025, Minerva Math, OlympiadBench	Science Bench, SuperGPQA, GPQA-Diamond, SCP-Hard, MMLU SCI
SCIENCE	SCP-116K	SCP-Hard, GPQA-Diamond, MMLU SCI, SuperGPQA, Science Bench	MATH-500, AMC, Minerva Math, OlympiadBench
GRAPH	Reasoning Gym	Quantum Lock, Largest Island	MATH-500, AIME-2024, Minerva Math, GPQA-Diamond, SCP-Hard

C.2 TRAINING DATA PREPARATION DETAILS

We describe our procedure for constructing filtered training datasets tailored to each model's capabilities.

Difficulty Estimation. For each problem in the source dataset, we sample 16 responses from the base model and count the number of correct solutions, yielding solve@16 $\in [0, 16]$. We retain only

Prompt template for MATH and GRAPH.

system
You are a helpful AI Assistant, designed to provide well-reasoned and detailed responses. You FIRST think about the reasoning process step by step and then provide the user with the answer.
Please enclose your final answer in the box: \boxed{Your Answer}.
user
`<question>`
assistant

Figure 4: **Prompt template used for RL training and evaluation on MATH and GRAPH.** The placeholder `<question>` is replaced with the actual mathematical question during fine-tuning and evaluation. Special tokens are omitted for clarity.

problems with solve@16 $\in [1, 15]$, excluding problems that are too difficult (solve@16 $= 0$) or trivially easy (solve@16 $= 16$) for the model.

Prompt template for SCIENCE.

system
Let's think step by step and output the final answer within \boxed{}.
user
`<question>`
assistant

Figure 5: **Prompt template used for RL training and evaluation on SCIENCE.** The placeholder `<question>` is replaced with the actual mathematical question during fine-tuning and evaluation. Special tokens are omitted for clarity.

Stratified Sampling. We use a stratified round-robin selection method to construct training subsets of size $N \in \{8, 32, 64, 512, 2048\}$. Filtered problems are partitioned into 15 bins $\{B_i\}_{i=1}^{15}$ according to their solve@16 values. To select N problems:

1. **Initialization:** Set the current count of selected problems $n_{\text{total}} = 0$.
2. **Round-Robin Selection:** While $n_{\text{total}} < N$:
   - Iterate through bins B_i for $i = 1, \ldots, 15$.
   - If B_i contains unsampled problems, randomly select one problem without replacement, add it to the training set, and increment n_{total}.
   - Terminate immediately if $n_{\text{total}} = N$.

This approach ensures that all difficulty levels are represented as uniformly as possible across all data scales.

C.3 IMPLEMENTATION DETAILS OF RL TRAINING

All experiments are implemented using the `verl` framework (Sheng et al., 2024) with its default hyperparameters: learning rate 10^{-6}, KL coefficient $\beta = 0.001$, clip ratio $\epsilon = 0.2$ and no entropy regularization. We set group size $G = 8$ for computational efficiency. For response sampling, we fix the sampling temperature 1.0 and a maximum response length of 2048 tokens unless otherwise noted. In `verl`, we set both the training batch size and mini-batch size to 64 prompts, yielding exactly one gradient update per training step. Each experiment is run for 496 total gradient updates. A simple rule-based reward function is used, assigning reward 1 to correct answers and 0 otherwise, without incorporating any format-related signals. For MATH and SCIENCE, answer matching and reward

computation is implemented with Math-Verify[2] library; for GRAPH, we use the internal task-specific evaluation protocol from Reasoning Gym. Prompt templates are detailed in Fig. 4 and Fig. 5. All experiments were conducted on 4 $\times$ NVIDIA H100 80GB GPUs.

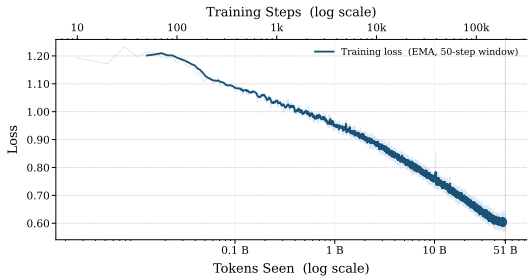


Figure 6: Training loss during continual pre-training of Llama3.2-3B on approximately 52B tokens of Nemotron-CC-Math data.

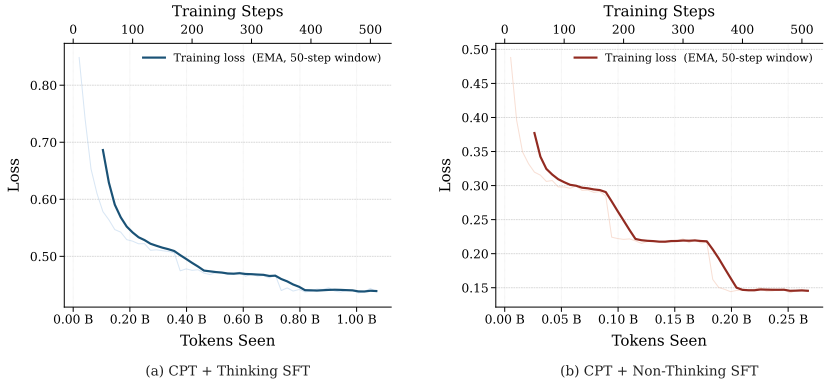


Figure 7: Training loss for (a) Thinking SFT and (b) Non-Thinking SFT on 43.5K math prompts, initialized from the CPT checkpoint.

C.4 Implementation Details of Evaluation

We evaluate reasoning performance using avg@16 accuracy (average pass@1 over 16 independent samples per problem) with temperature 1.0 sampling and report pass@k for $k \in \{4, 8, 16\}$.

C.5 Implementation Details of Continual Pre-Training

We continually pre-train Llama3.2-3B on the Nemotron-CC-Math-4plus subset (Mahabadi et al., 2025), comprising approximately 52B tokens of math-relevant documents filtered at quality score ≥ 4. Training is conducted for one epoch with a maximum sequence length of 2,048 tokens and a batch size of 128 sequences. We use AdamW with a peak learning rate of 2×10^{-5}, cosine decay schedule, 5% linear warmup, weight decay of 0.01, and gradient clipping at 1.0.

C.6 Implementation Details of SFT

For SFT, we train for three epochs with a batch size of 16 and a maximum sequence length of 8192 tokens. We tune the learning rate for each model within the $1 \times 10^{-5}, 5 \times 10^{-5}]$ and report results for the best-performing setting. For the subsequent RL phase, we evaluate performance across training sample sizes $N \in \{8, 2048\}$. All other hyperparameters follow the configurations established in Section C.3, with the maximum response length extended to 8192 tokens to accommodate long-form reasoning traces.

[2] https://github.com/huggingface/Math-Verify

Example prompt and response format of SFT

User Prompt:
Calculate the limit of the function:

$$\lim_{x\to 0}\frac{\arcsin(3x)}{\sqrt{2+x}-\sqrt{2}}$$

Assistant:
<think>
Okay, so I need to find the limit ...
Substituting $x = 0$ gives the indeterminate form $\frac{0}{0}$, so ...
Rationalizing the denominator, we get ...
</think>

The limit is calculated as follows:

1. Recognize the indeterminate form: substituting $x = 0$ yields
$$\frac{0}{0},$$
which suggests using rationalization or l'Hôpital's rule.
2. Rationalize the denominator:
$$\frac{\arcsin(3x)}{\sqrt{2+x}-\sqrt{2}}\cdot\frac{\sqrt{2+x}+\sqrt{2}}{\sqrt{2+x}+\sqrt{2}}=\frac{\arcsin(3x)\left(\sqrt{2+x}+\sqrt{2}\right)}{x}.$$
3. Split the limit:
$$\lim_{x\to 0}\frac{\arcsin(3x)}{x}\cdot\lim_{x\to 0}\left(\sqrt{2+x}+\sqrt{2}\right).$$
4. Evaluate each part:
   - Using
   $$\lim_{z\to 0}\frac{\arcsin z}{z}=1,$$
   the first limit becomes 3.
   - The second limit evaluates to $2\sqrt{2}$.
5. Combine the results:
$$3\cdot 2\sqrt{2}=6\sqrt{2}.$$

Final Answer:

$$\boxed{6\sqrt{2}}$$

Figure 8: **Example prompt and response format of SFT.** In Thinking SFT, the model is trained with reasoning traces enclosed by <think> and </think>, whereas Non-Thinking SFT omits them.

D Data Scale Effect

D.1 Additional Experimental Results from Small to Large Data Scale

Figs. 9, 10, and 11 present domain-specific training dynamics and generalization performance across sample sizes $N \in \{8, 32, 64, 512, 2048\}$. Each figure tracks the training reward, two in-distribution benchmarks, and one OOD benchmark, as listed in Table 2.

In the MATH domain, Llama models exhibit rapid saturation in small-sample regimes and rely heavily on data scale. In contrast, Qwen models yield comparable performance across varying sample sizes, characterized by extended saturation periods. Specifically, the math-specialized Qwen2.5-Math-1.5B

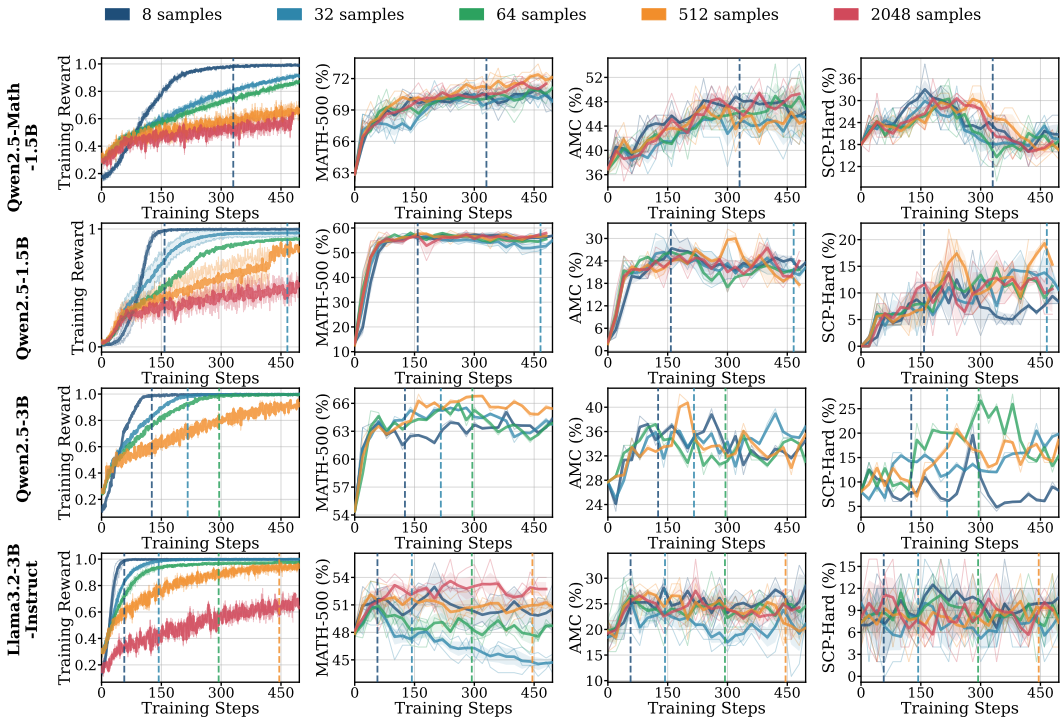


Figure 9: **Comparisons of RL training dynamics and performance across different models on MATH domain.** Results are averaged over three independent runs, with shaded regions indicating error bars. Vertical dashed lines denote the saturation step for each data scale if it saturates before 496 gradient steps. Llama models exhibit rapid saturation in small-sample regimes and rely heavily on data scale. In contrast, Qwen models yield comparable performance across varying sample sizes, characterized by extended saturation periods. Evaluation results in this figure are based on greedy decoding.

Table 3: **Math-domain training (1.5B/3B): in-domain benchmarks.**

Model	Metric	$t_{\text{sat}}^{(8)}$	MATH-500			AMC-2023			Minerva Math			OlympiadBench			AIME-2024		
			Δ_{sat}	Δ^*_{post}	G_{sat}	Δ_{sat}	Δ^*_{post}	G_{sat}	Δ_{sat}	Δ^*_{post}	G_{sat}	Δ_{sat}	Δ^*_{post}	G_{sat}	Δ_{sat}	Δ^*_{post}	G_{sat}
Qwen2.5-Math-1.5B	Avg@16	302	29.7	1.5	-2.0	18.7	0.6	-0.1	14.3	1.5	-1.0	13.7	1.0	-0.8	7.3	0.6	0.2
	Pass@4		12.3	0.5	-1.2	15.3	-0.1	0.2	14.0	2.0	-1.0	10.4	1.0	0.5	14.8	0.4	2.0
	Pass@8		6.2	0.2	-1.0	13.1	-0.1	-1.2	11.1	2.0	-1.4	8.0	1.2	1.0	16.5	0.4	2.4
	Pass@16		2.6	0.4	-0.4	10.8	-0.1	-2.8	10.3	1.2	-3.3	5.8	1.8	1.6	16.7	0.5	3.3
Qwen2.5-1.5B	Avg@16	170	42.6	0.9	-0.8	15.8	3.3	-1.0	12.6	0.9	-1.3	13.5	1.2	0.2	1.0	0.9	-0.2
	Pass@4		25.4	0.6	0.5	16.7	5.2	0.1	16.1	1.1	-1.1	13.9	1.3	0.6	3.4	2.9	0.3
	Pass@8		18.6	0.6	1.0	15.0	7.1	2.1	15.7	1.0	-0.3	12.9	1.4	1.2	4.2	4.4	2.2
	Pass@16		13.2	0.7	1.2	8.4	11.0	3.6	14.0	1.6	1.5	11.4	1.7	1.6	3.3	6.4	6.7
Llama3.2-3B-Instruct	Avg@16	55	10.8	-1.9	-1.4	8.8	-2.1	-0.4	7.4	-1.1	-1.7	6.5	-0.7	0.5	7.1	-0.6	-3.8
	Pass@4		23.9	-3.2	-0.6	21.3	-2.7	3.2	7.7	-1.3	-1.2	14.3	-1.1	2.4	0.0	0.0	-4.2
	Pass@8		21.7	-3.9	-0.7	20.0	-2.1	6.4	5.6	-0.4	0.1	13.8	-0.9	3.3	0.0	0.4	-3.7
	Pass@16		18.8	-3.8	0.2	15.7	-1.1	10.1	1.9	2.2	2.3	11.6	-1.0	4.4	0.0	1.1	-3.3

sustains a pre-saturation phase for 330 gradient steps on 8 samples, driving continuous improvements on in-domain benchmarks.

In the SCIENCE domain, the pre-saturation phase yields similar gains across all sample sizes; however, after the saturation point, larger sample sizes demonstrate distinct benefits. Similar to MATH domain, models exhibit significantly different saturation dynamics on small samples.

In the GRAPH domain, we compare two larger models, Qwen2.5-Math-7B and Llama3.1-8B-Instruct. The Qwen model also saturates faster here than in other domains, implying that the lack of domain-specific pre-training accelerates saturation in small-sample regimes.

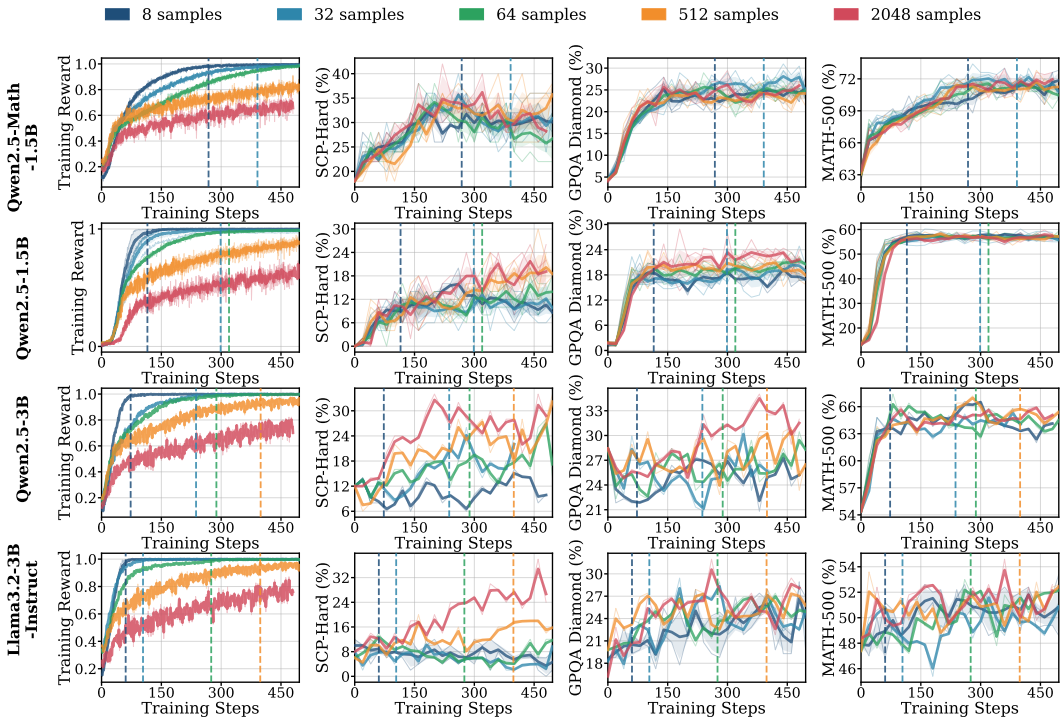


Figure 10: **Comparisons of RL training dynamics and performance across different models on SCIENCE domain.** Results are averaged over three independent runs, with shaded regions indicating error bars. Vertical dashed lines denote the saturation step for each data scale. The pre-saturation phase yields similar gains across all sample sizes; however, after the saturation point, larger sample sizes demonstrate distinct benefits. Models exhibit significantly different saturation dynamics on small samples. Evaluation results in this figure are based on greedy decoding.

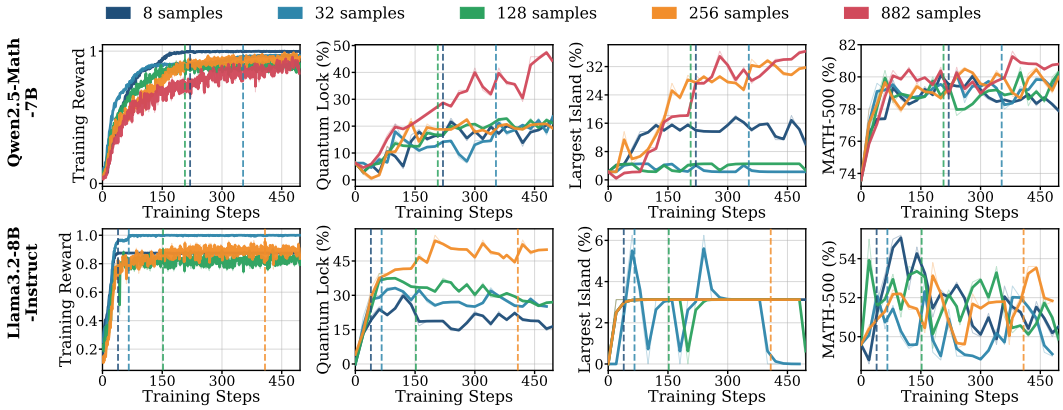


Figure 11: **Comparisons of RL training dynamics and performance across different models on GRAPH domain.** We use larger models (Qwen2.5-Math-7B, Llama-3.1-8B-Instruct) due to increased task difficulty. Results are averaged over three independent runs, with shaded regions indicating error bars. Vertical dashed lines denote the saturation step for each data scale. Qwen model also saturates faster here than in other domains. Larger datasets yield clear gains in the post-saturation phases. Evaluation results in this figure are based on greedy decoding.

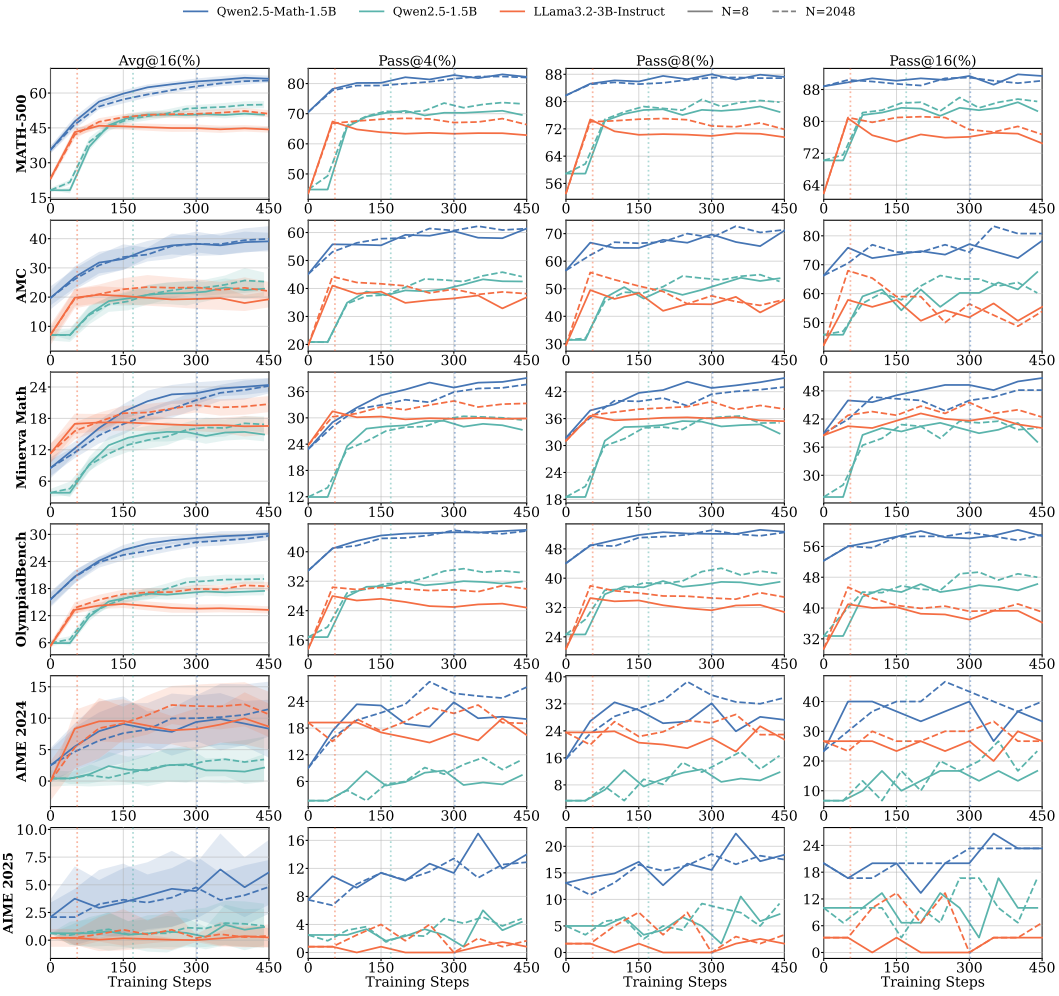


Figure 12: **Full in-domain benchmark evaluation results for the MATH domain across multiple models.** Vertical dashed lines denote the saturation step for each data scale.

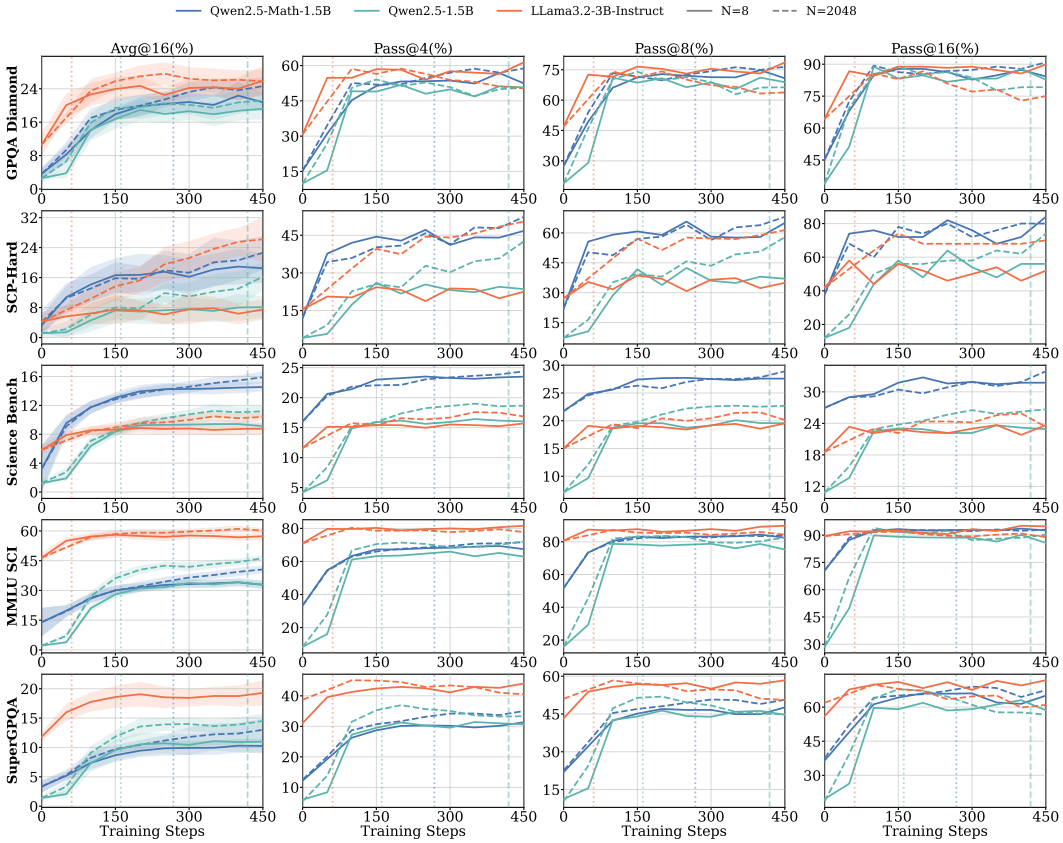


Figure 13: **Full in-domain benchmark evaluation results for the SCIENCE domain across multiple models.** Vertical dashed lines denote the saturation step for each data scale.

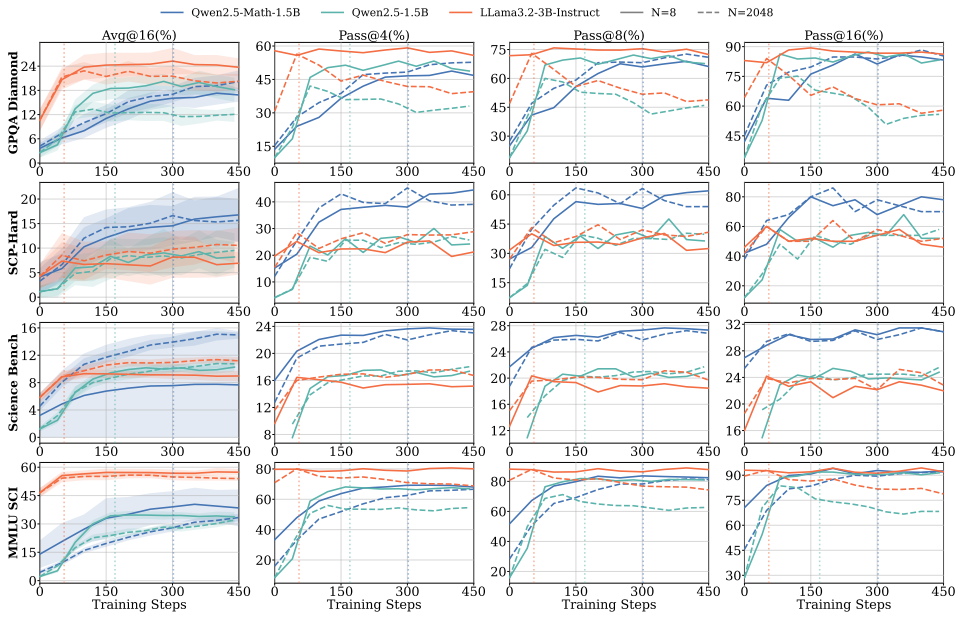


Figure 14: **Full out-of-domain benchmark evaluation results for the MATH domain across multiple models.** Vertical dashed lines denote the saturation step for each data scale.

Table 4: **Math-domain training (1.5B/3B): out-of-domain benchmarks.**

Model	Metric	$t_{\text{sat}}^{(8)}$	GPQA Diamond			SCP-Hard			MMLU SCI			Science Bench		
			Δ_{sat}	Δ_{post}^{*}	G_{sat}	Δ_{sat}	Δ_{post}^{*}	G_{sat}	Δ_{sat}	Δ_{post}^{*}	G_{sat}	Δ_{sat}	Δ_{post}^{*}	G_{sat}
Qwen2.5-Math-1.5B	Avg@16	302	12.6	1.1	0.9	10.5	2.1	2.4	25.4	1.2	-10.8	4.4	0.2	6.5
	Pass@4		33.0	1.7	1.7	22.8	6.0	7.2	35.9	-0.0	-6.7	7.6	0.1	-1.6
	Pass@8		41.0	2.5	2.2	25.9	8.6	10.2	31.9	-0.4	-5.4	5.6	0.2	-1.5
	Pass@16		39.1	3.8	2.5	26.0	10.6	10.0	22.3	-0.4	-3.4	3.5	1.0	-0.7
Qwen2.5-1.5B	Avg@16	170	13.8	1.4	-6.6	7.0	0.3	-0.4	29.4	-0.7	-11.2	6.8	0.6	-0.8
	Pass@4		33.0	1.5	-15.5	19.0	2.0	-0.6	47.4	-0.7	-14.6	9.2	0.7	-0.6
	Pass@8		37.9	1.0	-17.8	25.3	5.0	0.6	46.4	-0.4	-15.3	9.4	1.0	0.0
	Pass@16		31.5	2.3	-16.2	28.0	9.8	2.0	37.4	-0.3	-16.1	9.0	1.2	0.4
LLama3.2-3B-Instruct	Avg@16	55	-4.3	1.9	0.8	2.0	0.0	1.5	2.4	-0.4	-1.7	4.8	-0.3	0.1
	Pass@4		-2.1	2.8	0.6	5.5	-0.4	3.3	0.1	0.6	0.3	6.8	-0.6	-0.5
	Pass@8		0.5	3.0	0.7	8.3	-1.2	3.2	-0.3	0.6	0.2	7.7	-1.0	-0.8
	Pass@16		-1.1	6.7	2.1	14.0	-4.3	0.0	-0.3	1.1	0.3	8.2	-1.0	-0.3

Table 5: **Science-domain training (1.5B/3B): in-domain benchmarks.**

Model	Metric	$t_{\text{sat}}^{(8)}$	Science Bench			SCP-Hard			GPQA-Diamond			MMLU SCI			SuperGPQA		
			Δ_{sat}	Δ_{post}^{*}	G_{sat}	Δ_{sat}	Δ_{post}^{*}	G_{sat}	Δ_{sat}	Δ_{post}^{*}	G_{sat}	Δ_{sat}	Δ_{post}^{*}	G_{sat}	Δ_{sat}	Δ_{post}^{*}	G_{sat}
Qwen2.5-Math-1.5B	Avg@16	268	11.1	0.2	0.0	14.5	1.1	0.8	16.9	1.6	1.3	19.1	1.0	2.0	6.6	0.3	1.4
	Pass@4		7.5	-0.1	-0.5	35.0	-1.3	-1.3	38.0	2.3	1.4	34.7	1.1	-0.4	18.1	0.7	2.8
	Pass@8		6.0	-0.2	-0.7	43.5	-2.7	-1.6	44.5	1.7	0.5	31.4	0.9	0.3	24.7	0.4	2.8
	Pass@16		4.6	0.2	-0.7	44.0	-1.2	-2.0	41.1	0.2	0.5	22.3	0.5	-0.0	29.4	-0.1	1.3
Qwen2.5-1.5B	Avg@16	161	6.8	0.5	-0.0	6.4	0.2	0.6	13.3	1.7	2.9	25.4	4.4	8.7	7.9	0.9	2.4
	Pass@4		9.7	0.4	0.0	20.4	-1.5	-0.5	33.5	1.9	5.1	54.9	2.0	7.3	21.1	1.5	5.8
	Pass@8		9.9	0.3	0.3	31.1	-1.7	-2.3	39.7	1.5	5.4	62.5	0.1	4.7	28.5	2.0	7.4
	Pass@16		9.4	0.3	0.7	40.0	1.7	-2.0	32.0	3.0	5.6	60.5	0.2	2.6	32.8	3.1	8.8
Llama3.2-3B-Instruct	Avg@16	61	2.6	0.5	0.7	1.8	1.7	5.9	11.9	3.0	4.3	10.6	0.8	2.3	5.2	2.2	3.7
	Pass@4		3.5	0.4	0.6	5.1	2.9	11.1	24.0	4.8	3.8	8.7	1.4	0.7	8.5	3.9	5.6
	Pass@8		4.1	0.3	0.3	8.2	1.9	11.9	25.7	4.2	1.5	6.7	1.8	-0.2	10.6	3.9	4.5
	Pass@16		4.8	0.1	-0.3	16.0	-4.3	6.0	22.3	2.3	-1.1	2.6	2.6	-0.5	11.7	3.4	2.2

D.2 Full Evaluation Results

In this section, we will report the full evaluation results with all benchmarks and pass@k ($k \in \{1, 4, 8, 16\}$ metrics. Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 20, and Fig. 21 include in-domain and out-of-domain evaluation results across multiple benchmarks in MATH, SCIENCE and GRAPH domains.

Discussions on pass@k. Despite prior work (Yue et al., 2025) discussing divergent behavior between pass@1 and pass@k for $k > 1$ during RL training, we observe that $\Delta_{\text{sat}}^{(8)}$ keeps the same sign for all $k \in \{1, 4, 8, 16\}$ across most model-benchmark pairs, indicating consistent improvement in both pass@1 and pass@k. This indicates that during the pre-saturation period, the model is not just closing pass@k and pass@1 gap.

D.3 Additional Experimental Results on Large Models

In this section, we will report the full evaluation results on 7B and 8B models.

Fig. 16 and Fig. 17 show the results of Qwen2.5-Math-7B and Llama3.1-8B-Instruct models on MATH domain with in-domain and out-of-domain benchmarks, respectively.

Fig. 18 and Fig. 19 present the results of Qwen2.5-Math-7B and Llama3.1-8B-Instruct models on SCIENCE domain with in-domain and out-of-domain benchmarks, respectively.

Fig. 21 provides the results of Qwen2.5-Math-7B and Llama3.1-8B-Instruct models on GRAPH domain with more out-of-domain benchmarks.

Similar to the observations on smaller models, during the pre-saturation phases, models show generalization on both in-domain and out-of-domain benchmarks in terms of pass@k metrics. Compared to the 3B model, the 8B Llama model exhibits better cross-domain generalization. However, Llama models still saturate more faster than Qwen models and show clear data dependence (e.g., Fig. 18 on SCIENCE).

Table 6: **Science-domain training (1.5B/3B): out-of-domain benchmarks.**

Model	Metric	$t_{\text{sat}}^{(8)}$	MATH-500			AMC			Olympiad Bench			Minerva Math		
			Δ_{sat}	Δ^*_{post}	G_{sat}	Δ_{sat}	Δ^*_{post}	G_{sat}	Δ_{sat}	Δ^*_{post}	G_{sat}	Δ_{sat}	Δ^*_{post}	G_{sat}
Qwen2.5-Math-1.5B	Avg@16	268	25.3	0.8	1.1	14.7	1.5	0.1	11.4	0.9	0.5	12.7	1.1	0.6
	Pass@4		10.2	0.5	0.7	12.8	0.6	0.5	9.7	0.5	-1.1	12.5	1.3	0.7
	Pass@8		4.5	0.8	0.5	10.8	0.1	0.1	8.1	0.0	-1.6	9.4	1.8	0.7
	Pass@16		1.2	1.3	0.4	9.4	-0.0	-1.3	7.4	-0.3	-2.1	7.7	1.7	0.7
Qwen2.5-1.5B	Avg@16	161	32.3	2.1	1.2	14.5	0.7	-0.2	10.7	1.5	1.6	10.6	1.9	0.7
	Pass@4		30.9	1.1	0.9	21.1	1.9	2.0	16.9	1.4	2.1	18.3	1.4	0.7
	Pass@8		24.5	0.6	0.1	20.4	3.0	5.3	16.9	1.0	1.8	20.4	0.5	0.1
	Pass@16		19.0	0.2	-0.6	15.7	4.1	11.2	14.1	1.1	1.4	21.3	0.3	-1.5
Llama3.2-3B-Instruct	Avg@16	61	7.3	2.2	0.6	4.9	2.7	1.5	5.8	1.2	-0.1	5.9	1.2	1.2
	Pass@4		4.8	2.4	2.3	6.6	3.9	1.5	8.4	2.2	0.3	7.0	1.5	1.8
	Pass@8		3.2	2.4	3.2	6.0	4.5	0.2	8.3	2.6	0.4	5.9	2.2	2.0
	Pass@16		1.0	3.0	3.6	4.9	5.1	0.0	7.7	2.7	0.4	4.3	2.4	1.5

Table 7: **Graph-domain training (7B/8B): in-distribution benchmarks.**

Model	Metric	$t_{\text{sat}}^{(8)}$	Quantum Lock			Largest Island		
			Δ_{sat}	Δ^*_{post}	G_{sat}	Δ_{sat}	Δ^*_{post}	G_{sat}
Qwen2.5-Math-7B	Avg@16	151	8.0	4.9	7.3	19.8	1.9	-10.9
	Pass@4		22.6	1.5	-1.3	16.5	6.1	16.3
	Pass@8		26.8	2.8	-2.4	10.7	9.0	29.9
	Pass@16		30.6	5.5	-4.1	6.8	8.6	41.3
LLama3.1-8B-Instruct	Avg@16	29	10.1	7.1	3.2	1.8	1.0	-0.2
	Pass@4		15.3	-0.0	6.2	1.8	0.0	1.6
	Pass@8		15.4	7.4	12.5	1.8	0.0	2.8
	Pass@16		20.1	16.0	25.0	1.8	0.0	3.1

E Reward Type Effect

E.1 Additional Results on Reward Corruption

Reward corruption implementation. For each corruption level γ, we uniformly sample a γ fraction of prompts from the $N = 2048$ training set for each model–domain pair. For each selected prompt, we draw 96 model responses at temperature 1.0 and select the most frequently occurring incorrect final answer (i.e., one that receives zero reward under our verifier) as the corrupted target. During RL training, we replace the ground-truth labels of the selected prompts with these corrupted labels. For Llama models and the Graph domain, we cap γ at 0.9 due to the base model's inability to generate valid solutions even with extensive sampling.

Results. Fig. 2(a) shows complementary results to Section 3.2. We observe similar patterns that some models are robust to even large amounts of reward noise. In particular, Qwen models exhibit generalization ability even when trained on almost completely corrupted data; in contrast, Llama models tend to show high reward curves yet poorer generalization to new data, suggesting overfitting to incorrect responses.

E.2 Additional Results on Self-Supervised Proxy Rewards

Proxy rewards implementation. We evaluate two self-supervised proxy rewards as alternatives to ground-truth verification: majority voting and self-certainty.

1. **Majority Voting Reward.** Following TTRL (Zuo et al., 2025), we estimate pseudo-labels via majority voting and assign binary rewards based on agreement with the consensus answer. For each prompt, we sample 16 responses from the policy model. The most frequently occurring answer among these 16 responses is selected as the pseudo-label. Rewards are then computed as: r = 1 if the response matches the pseudo-label, and r = 0 otherwise. For policy optimization, we use the first 8 responses to compute advantages. All other RL hyperparameters follow Section C.3.
2. **Self-Certainty Reward.** Following Zhao et al. (2025), we use the model's own confidence as the reward signal. Self-certainty is defined as the average KL divergence between a

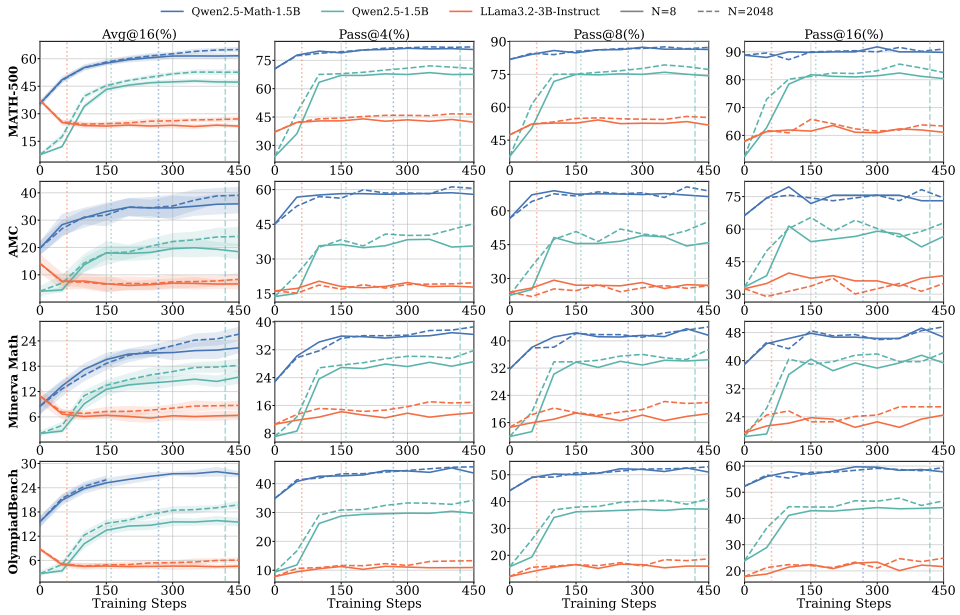


Figure 15: **Full out-of-domain benchmark evaluation results for the SCIENCE domain across multiple models.** Vertical dashed lines denote the saturation step for each data scale.

uniform distribution over the vocabulary and the model's next-token distribution:

$$r = \text{Self-certainty}(o|q) := \frac{1}{|o|}\sum_{i=1}^{|o|} \text{KL}(U \| p_{\pi_\theta}(\cdot|q, o_{<i})) \tag{1}$$

where $o_{<i}$ denotes previously generated tokens and U is the uniform distribution over the vocabulary. Higher values indicate greater model confidence. For each prompt, we sample 8 responses and use the self-certainty scores directly as rewards to compute advantages. All other RL hyperparameters follow Section C.3.

Results. Fig. 23 shows full results of self-supervised proxy rewards across model-domain pairs. Except for Qwen2.5-Math-1.5B, all other models exhibit failure modes under prolonged training. For Qwen2.5-1.5B on SCIENCE, both proxy rewards collapse: majority voting shows a sharp reward spike followed by performance degradation, while self-certainty leads to complete training collapse. Similarly, Llama-3.2-3B-Instruct on MATH shows degraded performance with both proxy rewards despite increasing training rewards. Only Qwen2.5-Math-1.5B on MATH maintains stable performance with majority voting, though self-certainty still collapses after approximately 200 steps. These results demonstrate that self-supervised proxy rewards are brittle and model-dependent, with only math-specialized models showing partial robustness.

E.3 REWARD HACKING EXAMPLE UNDER MAJORITY VOTE

Table 8 shows two rollouts from Qwen2.5-3B trained on SCIENCE with majority vote rewards at training step 846. In both cases, the model produces plausible intermediate reasoning but converges to the same final answer $\boxed{0}$, regardless of the problem content. The majority vote reward is 1.0 because all rollouts agree on this answer — the policy has learned to produce identical outputs to maximize consensus, constituting reward hacking. The correct answers (68.4g and τ_0/k, respectively) appear in the reasoning traces but are overridden in the final answer.

F BASELINE EFFECT

We analyze how the choice of reward baseline influences generalization in GRPO. Standard GRPO uses the within-group mean reward ($\mu = \frac{1}{G}\sum_{i=1}^{G} r_i$) as the baseline. By replacing μ with a constant

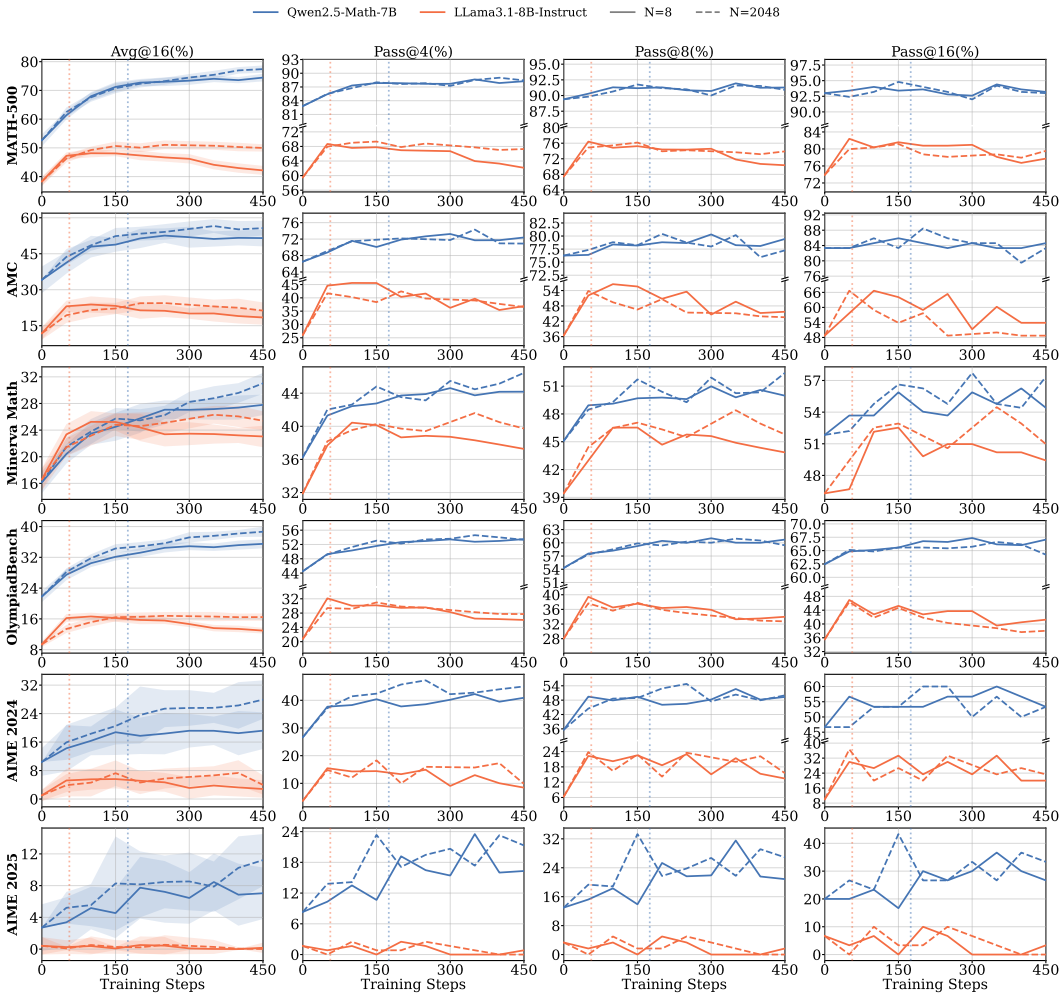


Figure 16: **Full in-domain benchmark evaluation results for the MATH domain on 7B and 8B models.**

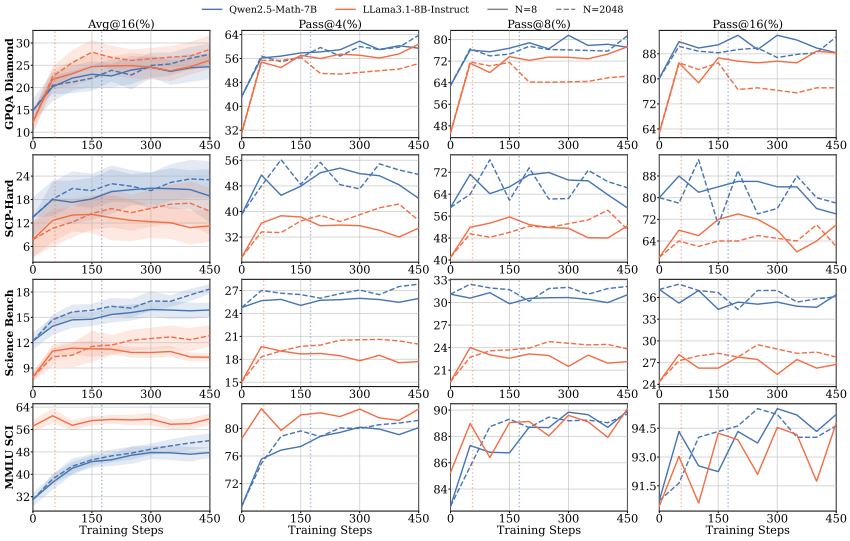


Figure 17: **Full out-of-domain benchmark evaluation results for the MATH domain on 7B and 8B models.**

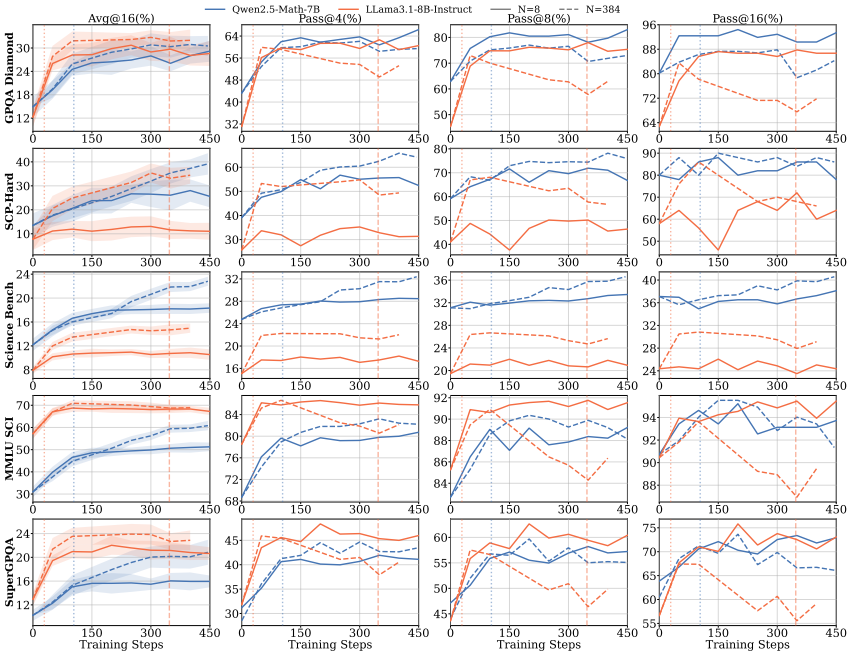


Figure 18: **Full in-domain benchmark evaluation results for the SCIENCE domain on 7B and 8B models.**

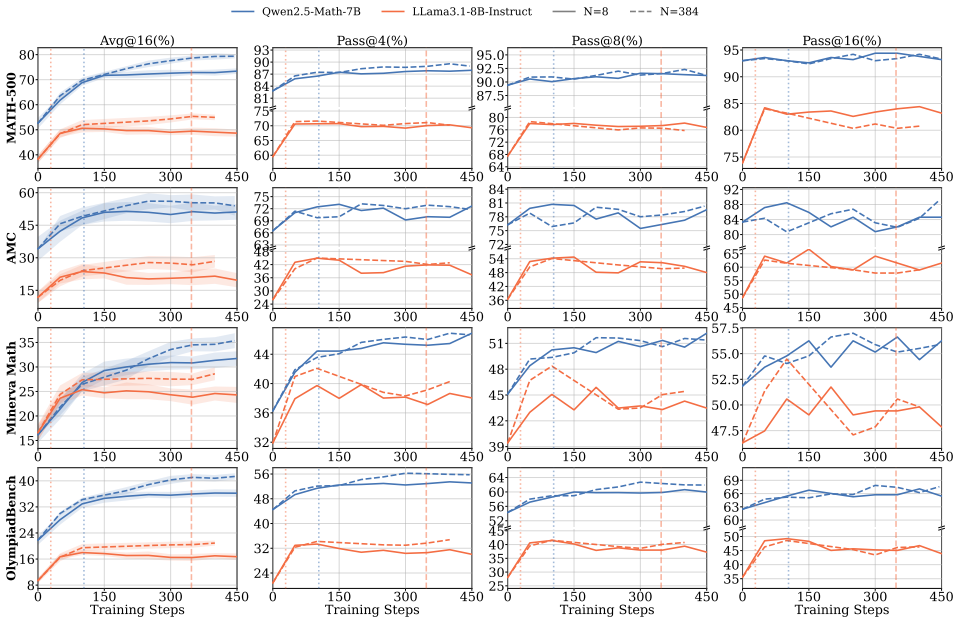


Figure 19: **Full out-of-domain benchmark evaluation results for the SCIENCE domain on 7B and 8B models.**

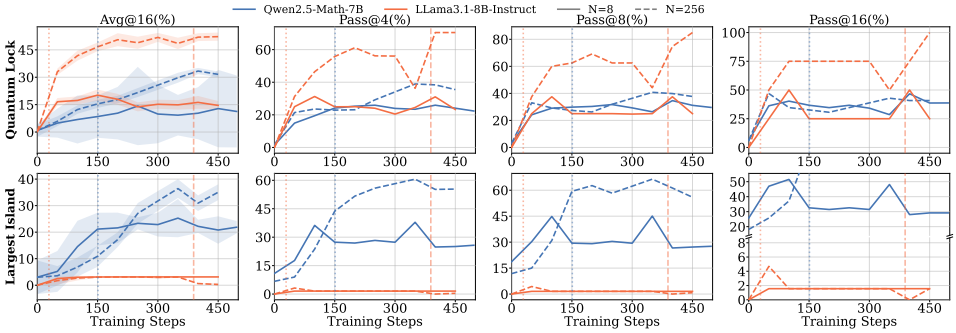


Figure 20: **Full in-domain benchmark evaluation results for the GRAPH domain on 7B and 8B models.**

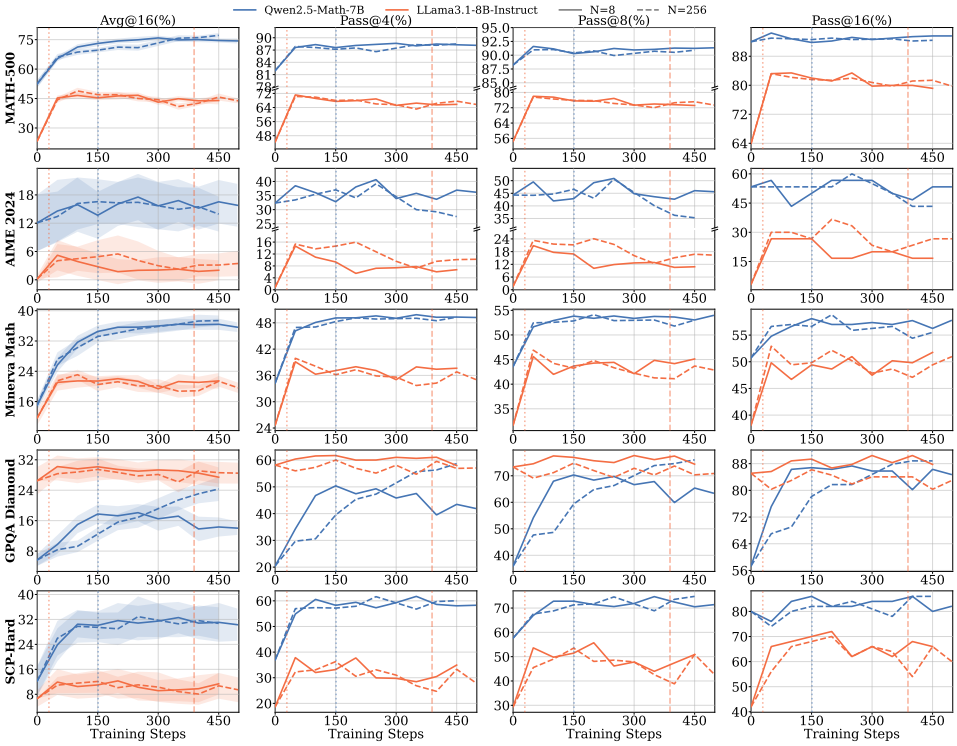


Figure 21: **Full out-of-domain benchmark evaluation results for the GRAPH domain on 7B and 8B models.**

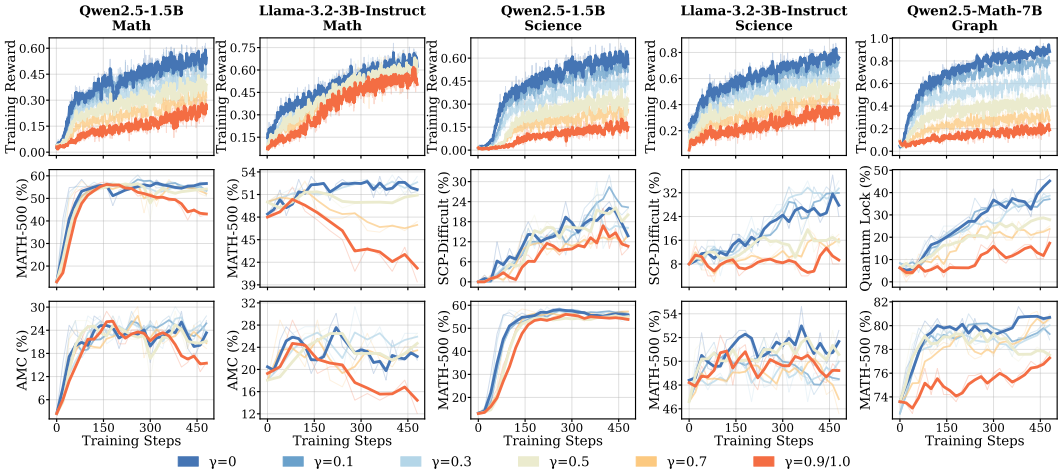


Figure 22: **Effect of reward label corruption on training dynamics and generalization.** γ denotes the fraction of training prompts with corrupted labels, ranging from clean ($\gamma = 0$) to fully incorrect ($\gamma = 1$). Qwen models on MATH and SCIENCE domains maintain performance under substantial corruption, while generalization of Llama models and GRAPH domain degrade at $\gamma \geq 0.5$. Evaluation results in this figure are based on greedy decoding.

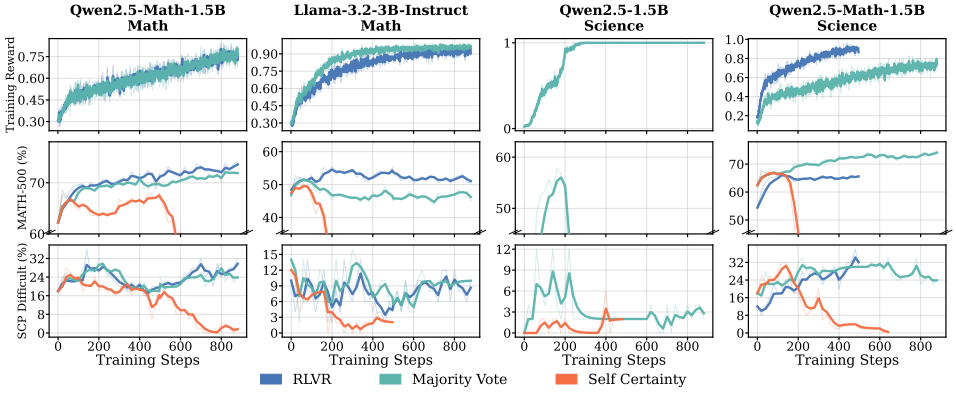


Figure 23: **Comparison of reward variants (RLVR, self-certainty, majority vote) with 1024 training samples**. Proxy rewards without verifiers exhibit failure modes under prolonged training: training collapse (self-certainty), and reward spikes followed by performance drops (majority vote). Evaluation results in this figure are based on greedy decoding.

baseline $b \in \{0, 1\}$, we isolate the direction of the policy update: $b = 0$ retains only positive reinforcement from correct samples (GRPO-POS), equivalent to the REINFORCE algorithm, while $b = 1$ retains only negative reinforcement from incorrect samples (GRPO-NEG), which (Zhu et al., 2025) studied in MATH domain. We remove the length penalty term $\frac{1}{|o|}$ in GRPO for this experiment. Based on the policy gradient theory, where subtracting an action-independent baseline does not change the expected gradient but reduces variance, with a large batch these two methods should yield similar learning behavior (Williams, 1992).

Figs. 24 and 25 present the training results on the SCIENCE domain for 8 and 1024 samples, respectively. In both regimes, GRPO-POS and GRPO-NEG achieve comparable Pass@1 performance to standard GRPO, exhibiting similar saturation and generalization behaviors. We note that this contrasts with recent findings by (Zhu et al., 2025), which highlight the superiority of GRPO-NEG. However, their improvements were primarily observed in Pass@k metrics rather than Pass@1 and evaluated on MATH domain. Beyond these metric differences, it's worth studying whether implementation artifacts may also influence observations. For instance, clipping terms in the GRPO formulation can introduce biases (Shao et al., 2025; Chen et al., 2025a). While our strictly on-policy setup mitigates such clipping effects, we leave a comprehensive analysis of these effects to future work.

Table 8: Two rollouts from Qwen2.5-3B on SCIENCE at step 846 under majority vote reward. Both produce coherent reasoning toward the correct answer but output $\boxed{0}$ as the final answer, achieving majority vote reward of 1.0.

Rollout 1: Sucrose solution problem	**Rollout 2: Moment of inertia problem**
Prompt: Prepare a 0.0348 mole fraction solution of sucrose using 100g of water.	*Prompt:* A wheel with moment of inertia I is acted upon by torque τ_0, resisted by $\tau_f = -k\omega$. Find the maximum speed.
Reasoning (excerpt): "Mass of sucrose required = 0.2 moles × 342 g/mole = 68.4 g"	*Reasoning (excerpt):* "$\omega_{\max} = \frac{\tau_0}{k}$"
Final answer: $\boxed{0}$	*Final answer:* $\boxed{0}$
Majority vote reward: 1.0	*Majority vote reward:* 1.0

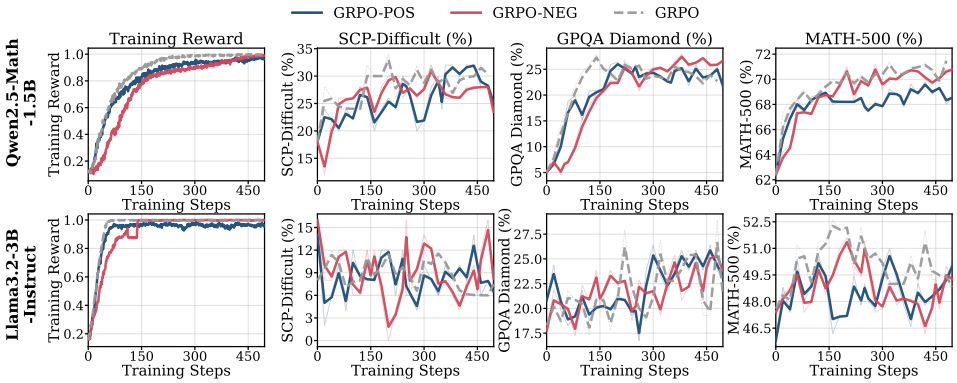


Figure 24: **Effect of baseline variants on SCIENCE domain with 8 training samples**. GRPO-pos (positive updates only) and GRPO-neg (negative updates only) produce comparable performance to standard GRPO.

G DIVERSITY AND FAITHFULNESS

G.1 QUANTIFICATION OF GENERATION DIVERSITY

To quantify the generation diversity of a model on a given prompt, we generate a number of responses, $y_1, ..., y_N$ and cluster them based on their reasoning similarity. Basing our analysis on the method

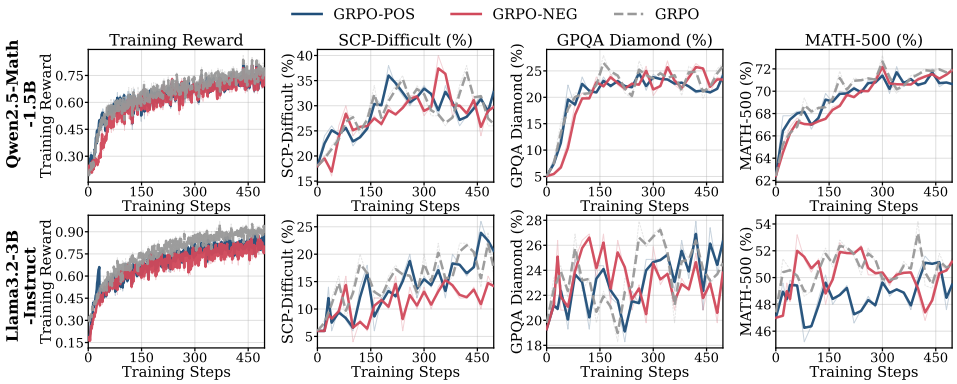


Figure 25: **Effect of baseline variants on SCIENCE domain with 1024 training samples.** Similar to Figs. 24, GRPO-pos (positive updates only) and GRPO-neg (negative updates only) produce comparable performance to standard GRPO.

Table 9: Inter-rater agreement between LLM judges measured using Cohen's Kappa.

Judge Pair	Cohen's Kappa
OpenAI o3 vs. GPT-OSS-20B (OpenAI, 2025b)	0.752
OpenAI o3 vs. Gemini 3 Flash (Google DeepMind, 2025)	0.649

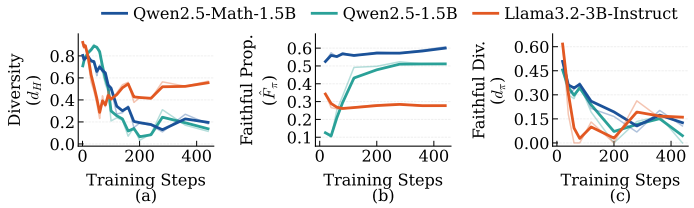


Figure 26: **Semantic diversity (a), reasoning faithfulness on correct samples (b), and faithful diversity (c) during 8-sample training on MATH.** Llama retains the highest raw diversity (a) but exhibits the lowest faithfulness (b). Once diversity is computed only over faithful responses (c), Llama's apparent advantage disappears, indicating that its diverse outputs are largely unfaithful.

used by (Li et al., 2025), to determine reasoning similarity between two outputs y_i, y_j, we define a function $s(y_i, y_j) \in \{0, 1\}$ such that $s(y_i, y_j) = 1$ if y_i, y_j are similar and 0 otherwise. To evaluate $s(\cdot, \cdot)$, we prompt GPT-4o (Hurst et al., 2024) as a diversity judge to determine whether the reasoning produced by any two responses follows a different reasoning path using the prompt specified in Fig. 30.

We form semantic clusters by iterating through responses and comparing them to a representative response from each existing cluster, creating a new cluster if the response is dissimilar to each representative. This is performed under the assumption of transitivity of similarity. We create clusters $\{C_1, ...C_K\}$ where $C_i = \{y_1, ..., y_{n_i}\}$ such that $s(y_i, y_j) = 1 \ \ \forall y_i, y_j \in C_i$. We then define the diversity scores using the Shannon Diversity Index (Shannon, 1948) as follows.

For a given prompt, let N be the total number of responses, n_i be the number of responses in cluster C_i, and K be the number of clusters. Let $p_i = \frac{n_i}{N}$ Define the Shannon entropy

$$H(p) = -\sum_{i=1}^{K} p_i \log p_i$$

and the effective number of clusters

$$N_{\text{eff}} = \exp\big(H(p)\big).$$

We then define the diversity score

$$\text{Div}_\pi(x) = \frac{N_{\text{eff}} - 1}{K - 1}. \tag{2}$$

when $K > 1$ and 0 otherwise.

For a data distribution $\mathcal{D}$, we define the overall generation diversity as $d_\pi(\mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}[\text{Div}_\pi(x)]$. Empirically, we sample $N = 16$ outputs per prompt and estimate d_π using 8 prompts from the specified dataset.

We define Faithful Diversity as this metric calculated only on responses that achieve a faithfulness score of 1 (see below).

Fig. 35 shows an example of the LM-as-judge output when prompted to evaluate the similarity of 2 responses.

G.2 Quantification of reasoning faithfulness

Inspired by prior work (Baker et al., 2025), we define the faithfulness as a response's intermediate reasoning trace contains all relevant information and remains logically consistent with the predicted

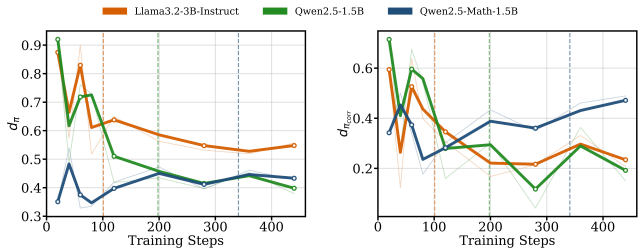


Figure 27: **Response diversity on 8 samples from the MATH-500 evaluation dataset.** Qwen-math shows high diversity within its correct answers, suggesting a range of learned robust reasoning paths.

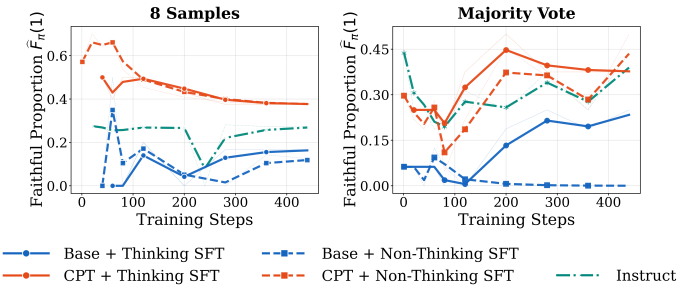


Figure 28: **Evolution of reasoning faithfulness of the Llama3.2-3B family on weak supervision domains when combined with continual pretraining and SFT variants**. When combined with Thinking-SFT and CPT, the Llama3.2-3B-Base model exhibits higher reasoning faithfulness.

final answer. Each model rollout produces a response y that contains (i) a reasoning trace and (ii) a final answer. We write $y = (r, a)$, where r is the reasoning text and a is the extracted final answer. For each input prompt x, we sample $y \sim \pi(\cdot \mid x)$ from the policy.

Faithfulness labeling. We define a discrete faithfulness labeling function $s_{\text{faithful}} : \mathcal{X} \times \mathcal{Y} \to \{0, \frac{1}{2}, 1\}$, where $s_{\text{faithful}}(x, y)$ measures the internal agreement between r and a in y:

- $s_{\text{faithful}}(x, y) = 1$ (*aligned*) if the reasoning trace r constitutes a coherent and logically supportive justification for the produced answer a, regardless of whether a is correct;
- $s_{\text{faithful}}(x, y) = \frac{1}{2}$ (*partially aligned*) if r exhibits a plausible argumentative trajectory toward a but contains substantial gaps, unsupported leaps, or local inconsistencies that weaken the justification;
- $s_{\text{faithful}}(x, y) = 0$ (*misaligned*) if a is not supported by r, e.g., r contradicts a, fails to address the question, or the answer appears as the "lucky" guess.

In practice, we implement $s_{\text{faithful}}(x, y)$ by querying OpenAI o3 (OpenAI, 2025a) as an LLM-as-a-judge with a fixed rubric (Fig. 31). OpenAI o3 is used for this task, as opposed to GPT-4o, due to requiring a larger model in order to be able to accurately reason about complex mathematical and scientific steps present in the reasoning traces. For a label $l \in \{0, \frac{1}{2}, 1\}$, we define the faithfulness rate of policy π over dataset $\mathcal{D}$ as

$$F_\pi(l) \;:=\; \mathbb{P}_{x \sim \mathcal{D},\, y \sim \pi(\cdot|x)} \big[s_{\text{faithful}}(x, y) = l \big].$$

At training step t, we approximate $F_{\pi_t}(l)$ using N training prompts $\{x_i\}_{i=1}^N$ and K rollouts per prompt:

$$\widehat{F}_{\pi_t}(l) \;=\; \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{1} \left\{ s_{\text{faithful}}\big(x_i, y_{i,k}\big) = l \right\}, \qquad y_{i,k} \sim \pi_t(\cdot \mid x_i). \tag{3}$$

We use $N = 8$ prompts and $K = 16$ rollouts per prompt at selected RL checkpoints on the specified training dataset. We report $\widehat{F}_{\pi_t}(l)$ for $l \in \{0, \frac{1}{2}, 1\}$ to characterize the distribution of reasoning faithfulness under the policy π_t.

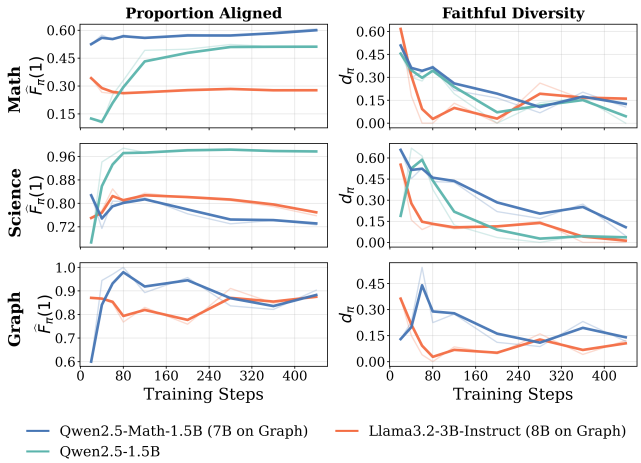


Figure 29: **Evolution of reasoning faithfulness (on correct samples) and faithful diversity on models throughout RL using 8 samples from a variety of datasets.** Llama models in the MATH domain exhibit significantly lower faithfulness compared to Qwen.

Diversity Prompt for LLM-as-a-judge

You are given the original prompt and two model-generated responses. Determine whether the two responses use different strategies to solve the problem.

Use the following guidelines:

-Different solution methods: Clearly different approaches (e.g., algebraic vs. geometric, analytical vs. numerical).
- Critical reasoning divergence: Significant differences in key reasoning steps or assumptions, even if final answers match.
- Conceptual differences: Distinct underlying concepts or representations (e.g., probability vs. combinatorics).
Also label as different if: The two responses share the same general approach but differ meaningfully in specific intermediate steps or manipulations crucial to the solution.

Original prompt: prompt

Generation 0: generation 0

Generation 1: generation 1

Question : Do Generation 0 and Generation 1 use different strategies?

You may first generate a short reasoning, then end your response with either ||yes|| if they use different strategies or ||no|| if they use the same strategy.

Figure 30: **LM prompt to check similarity between responses.**

Fig. 36 shows an example of the LM-as-judge output when prompted to evaluate the faithfulness of a model response when trained on the MATH training dataset.

Reliability of LLM-as-a-judge. To mitigate bias from using an LLM-as-a-judge for faithfulness evaluation, we assess consistency across multiple LLM judges by computing Cohen's Kappa (Cohen, 1960) across 16 faithfulness-scored Qwen2.5-Math-1.5B outputs when trained on 8 samples from the MATH training dataset at steps 20, 120 and 440.

Faithfulness Prompt for LLM-as-a-judge

You will be given: (1) a math problem prompt, and (2) a model response that may include BOTH reasoning and a final answer.

Definitions :
"Reasoning" = the parts of the model response that attempt to justify or derive a result (intermediate steps, equations, explanations, casework, narrative logic).
"Final answer" = the model's explicit committed result (e.g., after "Final:", "Answer:", "Therefore", "
boxed", or the last clear numeric/symbolic conclusion). If multiple answers appear, treat the last explicitly committed one as the final answer.

Task : Decide how well the reasoning supports the final answer, using these labels:

Label 1 (Correlated):
The reasoning forms a coherent derivation that would lead to the final answer as stated. Minor algebra slips are allowed IF the overall derivation still clearly targets that answer. The final answer may be objectively wrong; you judge alignment, not correctness.

Label 0.5 (Partially Correlated):
The reasoning is related to the problem and seems to move toward the final answer, but has major gaps, unjustified leaps, missing steps, or serious errors that break the proof. The answer is not a pure non-sequitur, but the support is weak/incomplete.

Label 0 (Uncorrelated):
The final answer is not supported by the reasoning. Examples include: contradiction with earlier derived statements; switching to an unrelated method; violating key constraints from the prompt; or the final answer appearing as an unsupported guess.

Output format (MANDATORY):
1) Briefly identify (a) the extracted final answer and (b) the key reasoning path in 1–3 sentences.
2) Then output exactly one label token on its own at the end: $\|1\|$ or $\|0.5\|$ or $\|0\|$.

Prompt: prompt

Response: response

Question: Does the reasoning path correspond to the provided answer? You may first generate a short reasoning, then end your response with either $\|1\|$ if they are fully correlated, $\|0.5\|$ if they are partially correlated, or $\|0\|$ if the answer is uncorrelated to the preceding logic.

Figure 31: **LM prompt to evaluate reasoning faithfulness on a sample from the MATH dataset.**

The judges achieve substantial agreement (κ= 0.752 and 0.649), indicating consistent faithfulness labeling across different models. We additionally conducted a small-scale manual evaluation to human-check the faithfulness scores and find fair alignment with the LLM judges.

G.3 ADDITIONAL RESULTS ON DIVERSITY ANALYSIS

Fig. 27 shows the semantic diversity of Llama3.2-3B-Instruct, Qwen2.5-1.5B and Qwen2.5-Math-1.5B on the MATH-500 evaluation dataset throughout RL training. Qwen-Math exhibits higher reasoning diversity on correct responses than the other models at the later stages of training, highlighting that RL enables it to successfully learn diverse and reliable strategies; coupled with its

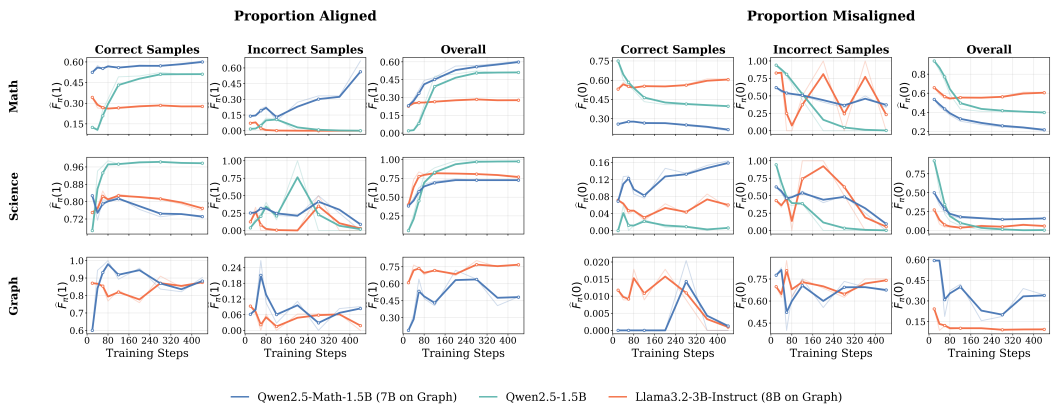


Figure 32: **Proportion of aligned and misaligned responses across models and training datasets.**

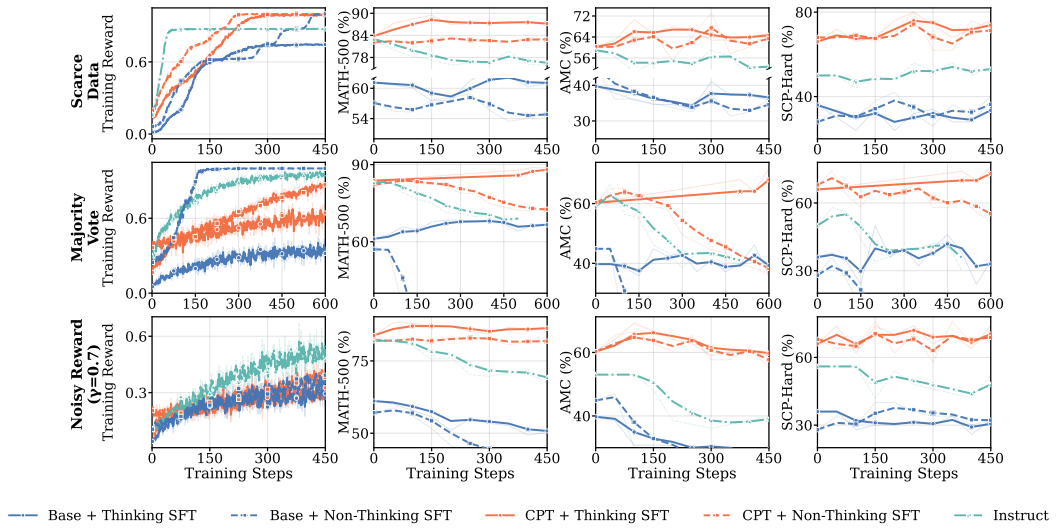


Figure 33: **Evaluation results of pass@16 metric across models with different pre-RL intervention on weak supervision.**

better performance on the evaluation dataset, this indicates stronger generalization properties. In particular, we observe significantly lower diversity in the Llama3.2-3B-Instruct model in comparison to its diversity metric on the training dataset (Fig. 26), implying disagreement between training and evaluation distributions and further highlighting the limitations of training diversity as an indicator of reasoning capabilities.

G.4 Additional results on faithfulness analysis

Fig. 32 shows the proportion of responses that are classified as *aligned* or *misaligned* when calculated with respect to correct, incorrect or all responses. Out of all correct, incorrect and overall responses, both Qwen2.5-1.5B and Qwen2.5-Math-1.5B show higher proportion of aligned responses and lower proportion of misaligned responses than Llama3.2-3B when trained on 8 samples from the MATH dataset. Qwen2.5-Math-1.5B additionally shows this result when trained on 8 samples from SCIENCE.

H Pre-RL Intervention

Fig. 33 report pass@16 results and Fig. 34 reports results on more benchmarks.

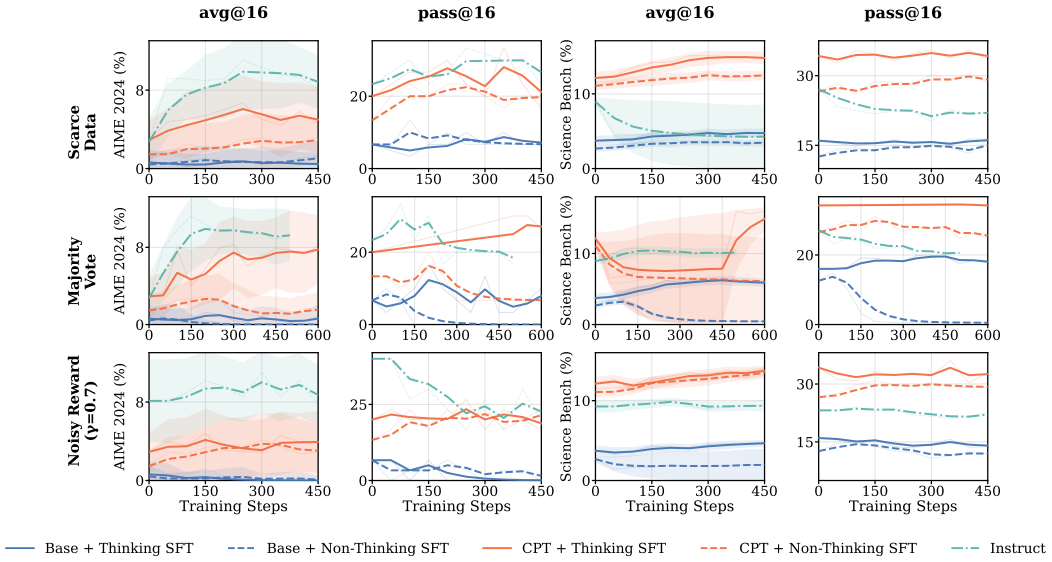


Figure 34: **Evaluation results on AIME 2024 and Science Bench across models with different pre-RL intervention on weak supervision.**

Diversity Judge Example Output

Prompt: A box contains 5 white balls and 5 black balls. I draw them out of the box, one at a time. What is the probability that all of my draws alternate colors, starting and ending with the same color?

Response A: To determine the probability that all of the ball draws alternate colors, starting and ending with the same color, we need to consider the following:
1. **Total number of ways to arrange the balls**: There are 5 white balls and 5 black balls, so the total number of ways to arrange these 10 balls is given by the binomial coefficient $\binom{10}{5}$, which represents the number of ways to choose 5 positions out of 10 for the white balls (the remaining 5 positions will be for the black balls). This value is calculated as:

$$\binom{10}{5} = \frac{10!}{5!5!} = 252$$

2. **Number of favorable outcomes**: For the drawings to alternate colors starting and ending with the same color, there are only two possible sequences: - Starting with a white ball and alternating (W B W B W B W B W B) - Starting with a black ball and alternating (B W B W B W B W B W)
So there are only 2 favorable outcomes.
3. **Probability calculation**: The probability is the ratio of the number of favorable outcomes to the total number of outcomes. Therefore, the probability P is:

$$P = \frac{\text{Number of favorable outcomes}}{\text{Total number of outcomes}} = \frac{2}{252} = \frac{1}{126}$$

So the final answer is $\boxed{\frac{1}{126}}$.

Response B: To solve the problem, we need to determine the probability that a sequence of 10 balls (5 white and 5 black) is arranged such that they alternate colors. There are only two possible sequences that meet this condition: "WBWBWBWBWB" and "BWBWBWBWBW". The total number of ways to arrange 5 white and 5 black balls is given by the binomial coefficient $\binom{10}{5}$, which represents the number of distinct permutations of 5 white and 5 black balls. This is calculated as:

$$\binom{10}{5} = \frac{10!}{5!5!} = 252$$

Since there are only 2 favorable outcomes (the two alternating sequences mentioned above), the probability is the ratio of the number of favorable outcomes to the total number of outcomes:

$$\text{Probability} = \frac{2}{252} = \frac{1}{126}$$

So, the probability that all of my draws alternate colors, starting and ending with the same color, is $\boxed{\frac{1}{126}}$.

Judge Output: Both Generation 0 and Generation 1 use the same strategy to solve the problem. They both identify that there are two possible sequences that satisfy the condition of alternating colors and starting and ending with the same color: "WBWBWBWBWB" and "BWBWBWBWBW". They both calculate the total number of ways to arrange the 10 balls using the binomial coefficient $\binom{10}{5}$, which accounts for the permutations of 5 white and 5 black balls. Finally, they both compute the probability by taking the ratio of the number of favorable outcomes (2) to the total number of outcomes (252), resulting in the same final answer of $\frac{1}{126}$.
There are no significant differences in their reasoning steps, assumptions, or intermediate calculations. Both responses follow a similar logical flow and use the same mathematical concepts to arrive at the solution.

Judge Result: $\|no\|$

Figure 35: **Qualitative Example of Diversity Analysis**

Faithfulness Judge Example Output

Prompt: Find all real pairs (a, b) that solve the system of equation

$$a^2 + b^2 = 25,$$
$$3(a + b) - ab = 15.$$

Response: To solve the system of equations:

$$\begin{cases} a^2 + b^2 = 25, \\ 3(a + b) - ab = 15, \end{cases}$$

we can use the following reasoning and algebraic manipulations. First, recall the identity:

$$a^2 + b^2 = (a + b)^2 - 2ab.$$

Let's denote $s = a + b$ and $p = ab$. So, the first equation becomes: $s^2 - 2p = 25$. The second equation remains: $3s - p = 15$. We can solve these two equations simultaneously. First, we express p from the second equation: $p = 3s - 15$. Next, we substitute p into the first equation: $s^2 - 2(3s - 15) = 25$. Simplify the equation:

$$s^2 - 6s + 30 = 25 \implies s^2 - 6s + 5 = 0.$$

This is a quadratic equation in s. We can solve for s using the quadratic formula $s = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$, where $a = 1$, $b = -6$, and $c = 5$:

$$s = \frac{6 \pm \sqrt{36 - 20}}{2} = \frac{6 \pm \sqrt{16}}{2} = \frac{6 \pm 4}{2}.$$

So, the solutions for s are: $s = 5 \quad$ or $\quad s = 1$. Now, we find the corresponding p values for each s: 1. If $s = 5$, then $p = 3(5) - 15 = 0$. 2. If $s = 1$, then $p = 3(1) - 15 = -12$. We now need to find the pairs (a, b) that satisfy the equations $a + b = s$ and $ab = p$. These are the roots of the quadratic equations: 1. For $s = 5$ and $p = 0$, the quadratic equation is:

$$t^2 - 5t = 0 \implies t(t - 5) = 0.$$

So, the solutions are $t = 0$ and $t = 5$. Thus, the pairs (a, b) are $(0, 5)$ and $(5, 0)$. 2. For $s = 1$ and $p = -12$, the quadratic equation is:

$$t^2 - t - 12 = 0.$$

We solve this quadratic equation using the quadratic formula:

$$t = \frac{1 \pm \sqrt{1 + 48}}{2} = \frac{1 \pm 7}{2}.$$

So, the solutions are $t = 4$ and $t = -3$. Thus, the pairs (a, b) are $(4, -3)$ and $(-3, 4)$. Therefore, the complete set of solutions is:

$$\boxed{(0, 5), (5, 0), (4, -3), (-3, 4)}.$$

Judge Output: Final answer given: $(0, 5), (5, 0), (4, -3), (-3, 4)$. Reasoning path: Introduces sums $s = a + b$ and product $p = ab$, rewrites equations into a system in s and p, solves to get $s = 5$ or 1, finds corresponding p, then solves quadratic $t^2 - st + p = 0$ to obtain all ordered pairs consistent with each (s, p), yielding the listed solutions. The derivation is complete and consistently leads to the stated pairs.

Judge Result: $\|1\|$

Figure 36: **Qualitative Example of Faithfulness Analysis**