WHEN DOES REASONING SFT GENERALIZE BETTER THAN GRPO?

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised fine-tuning (SFT) is often associated with memorization, whereas reinforcement learning (RL) is credited with stronger generalization in large language models (LLMs). We show that sufficiently trained SFT can achieve stronger out-of-distribution (OOD) generalization than Group Relative Policy Optimization (GRPO). Tracking SFT with long chain-of-thought (Long-CoT) supervision and GRPO over multiple epochs under matched training-problem sequences and checkpoints, we find that SFT surpasses GRPO across consecutive checkpoints after sufficient training, whereas GRPO is generally stronger or comparable at early checkpoints. Response analysis shows that strict verification language, identified by markers that directly check intermediate or final results, remains common in SFT OOD generations. The SFT–GRPO difference in verification density is already visible near GRPO's early high-performing checkpoint, before its later weakening in our matched run. Crucially, SFT's OOD advantage is not explained by simply generating longer reasoning: successful SFT responses are typically shorter and use verification more densely than failed responses. With further training, SFT's OOD performance declines from the stronger levels it reaches, although GRPO does not overtake SFT again within the observed training horizon. These results challenge the view that SFT memorizes while RL generalizes. SFT's generalization gains can be missed by stopping too early and eroded by training too long.

1 INTRODUCTION

Supervised fine-tuning (SFT) and Group Relative Policy Optimization (GRPO) are core reasoning-model post-training methods (Ho et al., 2023; Shao et al., 2024). The former learns from external reasoning responses, whereas the latter updates its policy using samples scored by verifiable rewards (Zelikman et al., 2022; Ho et al., 2023; Guo et al., 2025). Their generalization is often characterized differently: SFT is associated with memorization, whereas reinforcement learning is credited with stronger out-of-distribution (OOD) generalization (Chu et al., 2025). Yet whether this comparison persists throughout training remains unclear.

Prior OOD comparisons have not jointly examined complete long chain-of-thought (Long-CoT) supervision for SFT and sufficiently long multi-epoch training for both algorithms. Related studies vary in target format, training duration, and supervision (Chu et al., 2025; An et al., 2025; Xue et al., 2026; Ding et al., 2026; Kang et al., 2025; Kopiczko et al., 2026; Matsutani et al., 2026). These setting differences also make optimization progress a potential confound in SFT–GRPO comparisons. Because SFT and GRPO can progress differently under optimization, comparing them at a fixed early checkpoint can mix algorithmic differences with differences in optimization progress. If Reasoning SFT has not yet reached its stronger OOD performance at that checkpoint, the comparison can underestimate the performance SFT reaches with further training. The unresolved question is how the relative OOD performance of Reasoning SFT and GRPO changes when both methods are evaluated throughout sufficiently long multi-epoch training.

We show that SFT can generalize better. Tracking Long-CoT Reasoning SFT and GRPO across matched multi-epoch training, we find that their relative OOD performance changes markedly over training. At early checkpoints, GRPO is generally stronger or comparable to Reasoning SFT; after sufficient training, Reasoning SFT achieves higher OOD performance than GRPO across consecutive checkpoints, on both mathematical and cross-domain OOD benchmarks. Further training reduces

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.

SFT's OOD performance from the stronger levels it reaches, but GRPO does not overtake SFT again within the observed training horizon. Thus, stronger SFT generalization can be missed by stopping too early and eroded by training too long.

We compare SFT and GRPO response dynamics using a strict verification measure: language directly checking an intermediate or final result, excluding generic reflection or retry expressions. These markers are frequent in Long-CoT supervision and remain common in SFT OOD generations. The SFT–GRPO verification-density difference is already visible near GRPO's early high-performing checkpoint, before its later weakening in our matched run. Within SFT, successful responses are typically shorter and more verification-dense than failures. Reweighting SFT to GRPO's verification rate substantially narrows the observed OOD accuracy gap.

We further examine whether GRPO can produce a different OOD generation behavior beyond the differences associated with training dynamics. In a controlled state-transition game, GRPO amplifies an ID-rewarded, OOD-penalized strategy present in its training rollouts, achieving high ID accuracy while its OOD performance collapses after distribution shift.

Using these matched runs and complementary analyses, we obtain three main findings:

- **After sufficient multi-epoch training, Reasoning SFT achieves higher OOD performance than GRPO across consecutive checkpoints.** GRPO is generally stronger or comparable early in training; after sufficient training, Reasoning SFT surpasses GRPO across consecutive checkpoints and multiple OOD benchmarks. Further training reduces SFT's OOD performance from the stronger levels it reaches, but GRPO does not overtake SFT again within the observed training horizon.
- **SFT retains strict verification behavior on OOD problems.** The SFT–GRPO verification-density difference is already present at GRPO's highest-accuracy epoch on the fixed response-analysis set. Successful SFT responses are shorter and more verification-dense than failed responses.
- **GRPO may amplify an OOD-penalized strategy present in its training rollouts.** In our designed novel game, Reasoning SFT learns the complete and valid transition trajectories. In contrast, GRPO reinforces an OOD-penalized strategy, which achieves high training rewards and in-distribution (ID) accuracy. Moving to the OOD test, GRPO's accuracy collapses, while Reasoning SFT retains its performance.

2 PRELIMINARIES

In the main multi-epoch comparison, Reasoning SFT learns complete supervised responses (Ho et al., 2023); GRPO optimizes rewarded policy samples (Shao et al., 2024; Guo et al., 2025):

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{(x,y^S)\sim\mathcal{D}}\left[\frac{1}{|y^S|}\sum_{t=1}^{|y^S|}\log \pi_\theta(y_t^S \mid x, y_{<t}^S)\right],$$

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{x,z_{1:G}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|z_i|}\sum_{t=1}^{|z_i|}\ell_{\mathrm{clip}}(\rho_{i,t},\widehat{A}_i)\right].$$

Here x is a problem, y^S its supervised response, $\mathcal{D}$ their joint distribution, and π_θ the next-token policy with parameters θ. Indices i, t denote responses and tokens; $< t$ denotes preceding tokens and $|\cdot|$ token count. GRPO samples x from $\mathcal{D}$'s problem marginal and G responses $z_{1:G} = (z_1, \ldots, z_G)$ from rollout snapshot $\pi_{\mathrm{old}}(\cdot \mid x)$.

The policy ratio and PPO clipped surrogate (Schulman et al., 2017) are

$$\rho_{i,t} = \frac{\pi_\theta(z_{i,t} \mid x, z_{i,<t})}{\pi_{\mathrm{old}}(z_{i,t} \mid x, z_{i,<t})}, \qquad \ell_{\mathrm{clip}}(\rho_{i,t},\widehat{A}_i) = \min\Big\{\rho_{i,t}\widehat{A}_i,\ \mathrm{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\widehat{A}_i\Big\}.$$

With clipping width $\epsilon > 0$, clip restricts its first argument to $[1-\epsilon, 1+\epsilon]$.

Rewards combine final-answer correctness r_i^{cor} and format compliance r_i^{fmt} with respective weights $w_{\mathrm{corr}}, w_{\mathrm{fmt}}$: $r_i = w_{\mathrm{corr}} r_i^{\mathrm{cor}} + w_{\mathrm{fmt}} r_i^{\mathrm{fmt}}$. Each response shares advantage $\widehat{A}_i = (r_i - \bar{r})/\sigma_r$ across

Table 1: OOD performance at representative early and later checkpoints. Base is the shared step-0 evaluation. For each model–benchmark pair, the early checkpoint is the SFT avg10 minimum within steps 1–236 and the later checkpoint is the SFT avg10 maximum through step 708; both methods are reported at the same selected checkpoint.

Entry	Qwen3-8B				Qwen3-4B			
	AIME 2024	AIME 2025	GPQA D	MMLU Pro	AIME 2024	AIME 2025	GPQA D	MMLU Pro
Base	13.0/30.6	9.7/19.9	37.3/84.8	54.7/85.3	9.7/24.7	8.7/30.5	27.4/76.8	36.8/80.5
Early SFT minimum								
Checkpoint	8	2	16	16	8	8	88	8
SFT	8.0/27.9	8.3/27.3	29.3/72.9	28.0/75.5	7.7/25.9	4.0/**23.7**	23.4/54.4	**33.2/78.8**
GRPO	**12.3/28.5**	**9.7/27.9**	**37.4/83.6**	**52.4/82.7**	**10.0/27.3**	**6.7**/23.3	**23.7/57.8**	33.0/78.0
Later SFT maximum								
Checkpoint	590	472	590	708	708	590	472	472
SFT	**33.0/64.5**	**29.3/47.9**	**50.1/85.0**	**66.5/87.6**	**27.0/52.3**	**22.3/41.3**	**44.7/80.2**	**64.4/87.2**
GRPO	14.3/33.3	11.3/25.9	38.0/81.4	55.2/81.1	8.3/21.3	4.0/16.7	37.3/66.7	50.3/76.7

tokens, where $\bar{r}, \sigma_r$ are the group reward mean and standard deviation; $\widehat{A}_i = 0$ when $\sigma_r = 0$. Both methods use one response per GPU microbatch, so `token-mean` averages that response's valid tokens, consistent with the normalization above (Appendix A.2).

3 SFT–GRPO OOD Performance Across Training

A single checkpoint can conflate algorithmic differences with optimization progress. We therefore track matched multi-epoch trajectories: GRPO is often stronger or comparable early, sufficient training reveals an SFT OOD advantage across consecutive checkpoints, and further training reduces SFT's OOD performance from the stronger levels it reaches.

3.1 Experimental Setup

Matched multi-epoch comparison. Both methods start from the same Base model and follow the same sequence of 7,500 problems for six epochs (708 optimizer steps). They therefore repeatedly traverse the same finite problem set. This design matches problem exposure, problem order, and optimizer-update checkpoints across SFT and GRPO, but does not select an individually optimal stopping point for either method. In the main GRPO run, each problem uses eight rollouts. Figure 1 reports the trajectories; Table 1 summarizes early and later checkpoints. Complete optimization settings are in Appendix A.2.

Models and training data. The main-text comparison uses Qwen3-8B-Base and Qwen3-4B-Base (Yang et al., 2025). Both methods train on the same 7,500 problems selected from OpenR1-Math-220k (Hugging Face, 2025). SFT receives complete Long-CoT supervision; GRPO uses verifiable final-answer rewards on policy rollouts. Details are in Appendix A.1.

Benchmarks and evaluation protocol. We evaluate AIME 2024/2025 and transfer beyond the mathematical training domain on GPQA-D (Rein et al., 2023) and MMLU-Pro (Wang et al., 2024). From ten responses per problem and checkpoint, we report avg10 and pass@8 (Chen et al., 2021). Prompts, decoding settings, and seeds are fixed across methods and checkpoints (Appendix A.4).

3.2 Sufficient Training Reveals an SFT OOD Advantage

Figure 1 shows GRPO stronger or comparable at many early checkpoints, followed by a sustained SFT advantage across consecutive later checkpoints. Because this separation appears at different times across benchmarks, fixed early training budgets can underestimate sufficiently trained SFT.



Figure 1: OOD trajectories on Qwen3-8B (top) and Qwen3-4B (bottom). Color identifies the method; open circles and filled squares denote avg10 and pass@8. The horizontal axis uses actual optimizer steps; 118–708 mark epochs 1–6. Both methods share the Base evaluation at step 0. This shows Reasoning SFT OOD performance is better than GRPO after sufficient training.

Table 1 summarizes this change at SFT-selected early and later checkpoints, with both methods evaluated at the same steps; Appendix B.1 provides all results.

Mathematical reasoning. At the representative later checkpoints, Qwen3-8B SFT reaches 33.0%/64.5% avg10/pass@8 on AIME 2024 and 29.3%/47.9% on AIME 2025, versus GRPO's 14.3%/33.3% and 11.3%/25.9%. Qwen3-4B shows the same later advantage: SFT avg10 reaches 27.0% and 22.3%, versus 8.3% and 4.0% for GRPO.

Transfer beyond mathematics. On GPQA-D, later SFT avg10 reaches 50.1% on Qwen3-8B and 44.7% on Qwen3-4B, versus GRPO's 38.0% and 37.3%. On MMLU-Pro, the corresponding values are 66.5% and 64.4% for SFT, versus 55.2% and 50.3% for GRPO. Table 1 shows the same ordering for pass@8. The later SFT advantage also holds on Olmo3-7B (Team Olmo, 2025) (Appendix B.1–B.2).

The advantage also appears when SFT uses the same 10^{-6} peak learning rate as GRPO: at step 708, SFT reaches 25.0% avg10 on AIME 2024 and 64.6% on MMLU-Pro, versus 15.3% and 55.2% for GRPO (Appendix B.3). Equal nominal rates do not imply equal effective updates or establish GRPO's optimal learning rate.

The advantage is also clear at a fixed checkpoint. Table 2 fixes Qwen3-8B SFT at step 708 and compares it with GRPO's highest observed avg10 over steps 1–708. SFT is higher on all four benchmarks, by 2.2–12.7 percentage points. The highest observed GRPO checkpoint is early on each benchmark (Table 2), so this comparison does not rely on selecting only a late GRPO checkpoint.

Table 2: Qwen3-8B SFT at step 708 versus GRPO's highest observed avg10 over steps 1–708. Scores are percentages; differences are percentage points.

Benchmark	SFT step 708	GRPO highest observed	GRPO step	SFT − GRPO
AIME 2024	**32.7**	20.0	236	+12.7
AIME 2025	**28.0**	18.0	147	+10.0
GPQA-D	**48.8**	46.6	236	+2.2
MMLU-Pro	**66.5**	58.0	64	+8.5

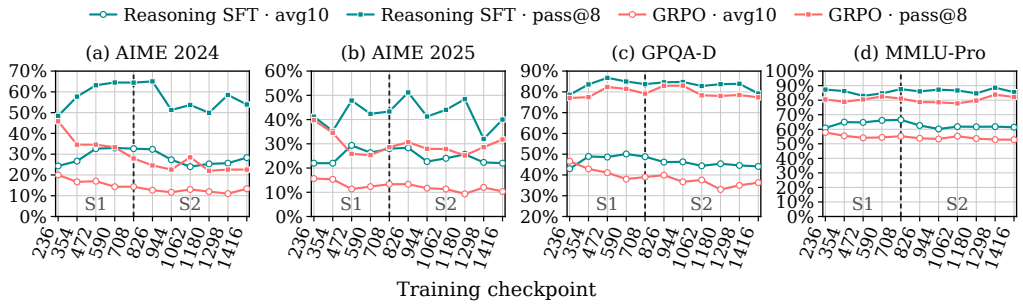


Figure 2: Qwen3-8B OOD trajectories through twelve epochs. The dashed line at step 708 separates the primary comparison from further training; both methods are evaluated through step 1,416. With further training, SFT's OOD advantage weakens.

3.3 Further Training Weakens SFT OOD Performance

We extend both Qwen3-8B methods from six to twelve epochs, ending at step 1,416, and evaluate matched epoch checkpoints (Figure 2). This extension tests whether SFT continues to improve with additional exposure to the same training problems and whether it remains ahead of GRPO.

Longer training reduces SFT performance across benchmarks. From step 708 to step 1,416, SFT avg10 falls by 4.3, 6.0, 4.7, and 5.1 percentage points on AIME 2024, AIME 2025, GPQA-D, and MMLU-Pro, respectively, while pass@8 also ends lower on all four benchmarks. The decline therefore appears in both mathematical and cross-domain OOD evaluation.

The decline is not monotonic, and avg10 and pass@8 can peak at different later checkpoints (Appendix B.2.1).

Despite this decline, SFT remains ahead of GRPO at step 1,416 on both avg10 and pass@8 across all four benchmarks, with avg10 margins ranging from 7.8 to 15.0 percentage points. Thus, further training weakens SFT's previously stronger OOD performance, but GRPO does not overtake SFT within the observed horizon. Full trajectories and learning-rate sensitivity are in Appendices B.1–B.3; Section 4 examines the accompanying response changes.

4 Response Dynamics of SFT and GRPO

We examine response changes accompanying SFT's OOD advantage (Section 3) with a strict verification measure. Markers common in Long-CoT supervision remain frequent in SFT OOD generations; the density difference from GRPO already appears near its early peak on the fixed response-analysis set. SFT failures are longer, less verification-dense, and more frequently length-limited than successes.

4.1 Experimental Setup

Our strict verification measure requires language directly checking an intermediate or final result. Markers cover *verify*, *confirm*, and *validate*; double-, cross-, and sanity-checking; substituting results back; numerical, unit, dimensional, boundary, and special-case checks; and testing candidate values, solutions, or answers. Generic reflection or retry language—*wait*, *rethink*, or *try again*—and standalone *check* are not sufficient.

Verification rate is the fraction of responses containing at least one strict verification marker. **Verification density** is the number of strict-marker occurrences per 1,000 generated response tokens, computed per response before averaging. Response length excludes prompt tokens. A response is **length-limited** when its stored termination reason is `max_tokens`; a **normal stop** has `finish_reason=stop`.

At step 708, we compare verification rates and SFT accuracy with/without verification on the four OOD benchmarks. Epochs 1–6 use fixed problems and shared sampling seeds; intervals use problem-

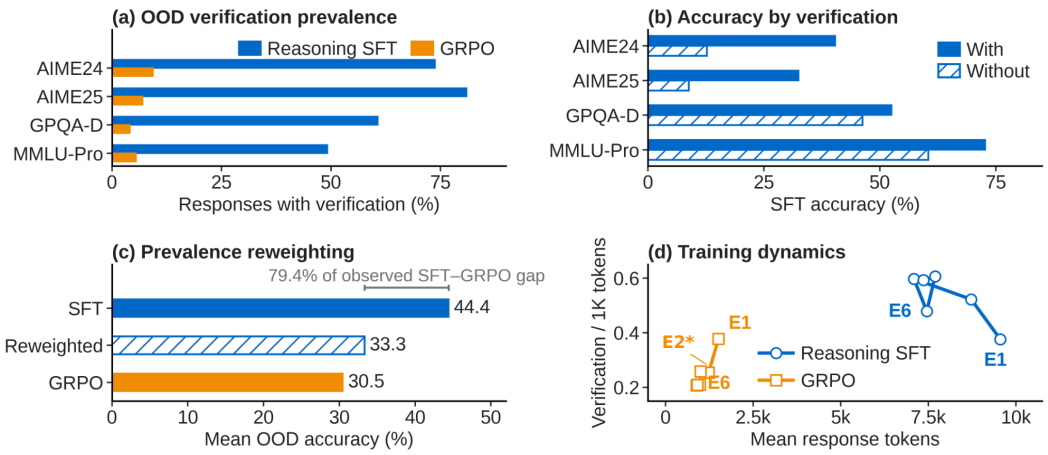


Figure 3: Verification in OOD responses and training dynamics. (a) Step-708 OOD strict-verification rates. (b) SFT accuracy with and without strict verification. (c) Mean OOD accuracy for observed SFT, SFT reweighted to GRPO's verification rate with conditional accuracies fixed, and observed GRPO. (d) Mean response length and verification density over epochs 1–6. E1 and E6 mark the endpoints; E2* marks the highest-accuracy GRPO epoch on the fixed response-analysis set. Appendix C reports definitions, intervals, and additional analyses.

level bootstrap resampling (Appendix C). GRPO's highest accuracy on this fixed response-analysis set occurs at epoch 2, which serves as an early reference alongside checkpoint-matched comparisons.

4.2 SFT Transfers Verification to OOD

At least one strict verification marker appears in 92.24% of the Long-CoT responses used for SFT. At the matched step-708 checkpoint, the corresponding rate remains 49.2–81.0% across the four SFT OOD benchmarks, compared with 4.1–9.3% under GRPO. Verification thus remains common on unseen problems, including benchmarks outside the mathematical training domain.

On every benchmark, SFT responses containing verification are more accurate than those without it. The differences are **27.7, 23.7, 6.3, and 12.3 percentage points**, respectively.

For **verification-rate reweighting**, we keep SFT accuracy with and without verification fixed, but replace SFT's verification rate with GRPO's corresponding rate. Mean SFT accuracy falls from **44.42% to 33.34%**, compared with observed GRPO accuracy of **30.47%**. This reduction corresponds to **79.4% of the mean observed SFT–GRPO accuracy gap**.

4.3 SFT and GRPO Fail Differently

On the fixed response-analysis set, GRPO reaches its highest accuracy at epoch 2 (58.7%). At this checkpoint, verification density is 0.522 per 1K tokens for SFT and 0.254 for GRPO, a paired difference of $+0.268$ [0.138, 0.408]. From epoch 2 to epoch 6, GRPO accuracy decreases to 54.3%, mean response length decreases from 1,231 to 911 tokens, and its strict verification rate decreases from 21.9% to 16.3%. The density difference therefore precedes this later GRPO decline.

Longer Long-CoT is not itself better. Successful SFT responses typically finish in **5.6k–6.8k tokens**, whereas failures extend to **13.6k–15.3k tokens**. Across epochs, **51–87%** of failed SFT responses reach the 16,384-token limit. Successful responses consistently have higher verification density. The longest SFT generations are its failures, not its successes.

These patterns associate SFT's advantage with denser verification and timely termination, rather than generation length alone.

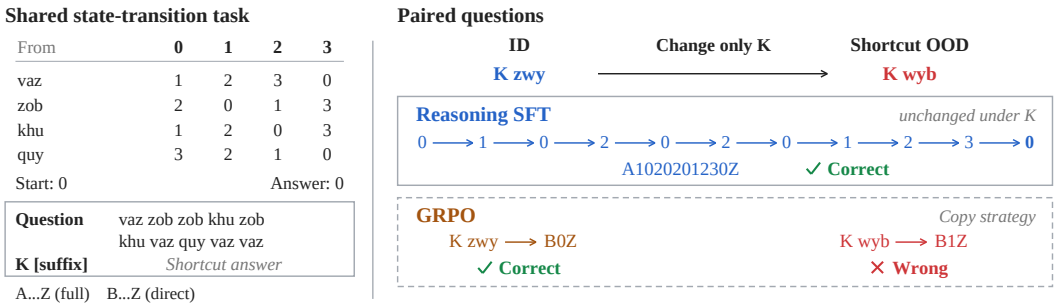


Figure 4: State-transition task and paired Shortcut-OOD design. Each problem requires applying a transition table to an operation sequence, while the suffix K provides an ID shortcut to the final state. Shortcut OOD changes only K, leaving the transition trajectory and correct answer unchanged, so the full-trajectory strategy remains valid while the copy strategy becomes incorrect.

5 GRPO Amplifies an ID-Rewarded Shortcut and Collapses Under Paired OOD

Outcome rewards identify correctness without uniquely specifying the response that produces it. We use a controlled state-transition game to test whether GRPO amplifies responses that follow an ID-rewarded shortcut and whether this leads to OOD collapse.

5.1 The Novel State-Transition Game and Paired Shortcut OOD Design

The game contains four states $\mathcal{S} = \{0, 1, 2, 3\}$ and four symbolic operations $\mathcal{O} = \{\texttt{vaz}, \texttt{zob}, \texttt{khu}, \texttt{quy}\}$. Each problem specifies a transition table T, an initial state s_0, and an operation sequence $o_{1:L}$, with $s_t = T(o_t, s_{t-1})$.

Every valid output begins with `A` or `B` and ends with `Z`. `A`$s_1 \cdots s_L$`Z` gives the complete transition trajectory, whereas `B`s_L`Z` gives only the final state; both receive the same reward when correct. Each prompt also contains a suffix K mapped by a fixed codebook κ to a state. Under Shortcut ID, $\kappa(K) = s_L$, allowing the model to bypass the transitions and output `B`$\kappa(K)$`Z`.

Paired Shortcut OOD intervenes on K, changing its code from s_L to $(s_L + 1) \bmod 4$ while fixing the transition table, initial state, operations, correct trajectory, and answer. This tests whether outputs follow K when it conflicts with the answer. The no-shortcut control retains K and κ but sets $\kappa(K) \neq s_L$ on every training example: copying never earns correctness reward. Arbitrary symbols and the fixed `A`/`B`...`Z` format reduce task-specific pretraining and formatting effects. Figure 4 shows a pair.

5.2 Experimental Setup

All experiments start from Qwen3-8B-Base. We use GRPO (Shao et al., 2024; Guo et al., 2025) with a task-specific final-answer reward: correct `A`- and `B`-responses receive equal reward. SFT supervises complete `A`-trajectories as a learnability reference, rather than a matched algorithmic comparison. Training, evaluation, and SFT-warmup, process-reward, and shortcut-strength controls are detailed in Appendices A.3, D.2, and D.3.

5.3 GRPO Amplifies the Rewarded Shortcut and Collapses Under Paired OOD

At checkpoint 64, GRPO reaches 100.0%/0.0% Shortcut-ID/OOD accuracy, versus SFT's 94.43%/94.43% (Figure 5). Since only K changes within pairs, the gap measures suffix sensitivity with the transition problem fixed. The large GRPO ID gain disappears when direct copying is always incorrect during training.

The B-format copy rate rises from 5.1%/4.0% on Shortcut ID/OOD in Base samples to 79.5%/81.0% after GRPO (Appendix D.1). OOD copies follow the incorrect K; combined with the paired

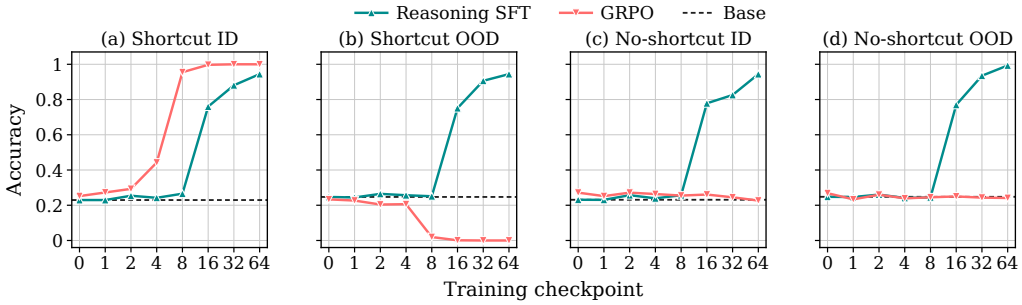


Figure 5: Accuracy across the shortcut and no-shortcut conditions. Panels (a–b) show Shortcut ID and paired Shortcut OOD; panels (c–d) show no-shortcut ID and OOD evaluations. Dashed horizontal lines mark the Base evaluations used for the SFT trajectories.

intervention, this supports K-dependent answer selection. Base matches establish only that copy-compatible outputs are sampleable.

Correct A- and B-format responses receive the same outcome reward, although the canonical B-format copy response is shorter. The present comparisons establish shortcut dependence without isolating a causal effect of response length.

With four-update SFT warmup, the implemented local process reward, or reduced training-K correctness, additional runs retain the ID–OOD split (Appendices D.2 and D.3). After SFT-4, the final-answer K-following rates at $g = 64$ are 95.2% and 98.7% under outcome-only and process-reward GRPO, despite almost no explicit B-format copying. Sixteen-update warmup instead preserves high ID and OOD accuracy. Thus, K dependence can occur across formats, and the warm-up outcomes differ with initialization.

Overall, these runs show that GRPO can reinforce responses that obtain ID reward by following K, while their performance collapses under a controlled change to K.

6 Related Work

6.1 SFT and RLVR

SFT learns from reasoning demonstrations (Ho et al., 2023), whereas RLVR optimizes policy samples using outcome rewards (Luong et al., 2024; Shao et al., 2024; Guo et al., 2025). OOD comparisons vary across tasks and settings: SFT can fit its supervision more closely, while RL can transfer better to unseen variants (Chu et al., 2025). At the reasoning-trajectory level, SFT has been reported to expand correct trajectories, whereas RL compresses incorrect trajectories (Matsutani et al., 2026). Existing comparisons also differ in supervision, training duration, initialization, and evaluation.

6.2 SFT Training Dynamics

Candidate-response probabilities evolve during finetuning (Ren & Sutherland, 2025). Long-CoT SFT can exhibit OOD dip-and-recovery dynamics (Ren et al., 2026) and has also been studied through understanding-to-reasoning transitions (An et al., 2025). Under a fixed update budget, repeating complete solutions can outperform using more distinct examples (Kopiczko et al., 2026). Long-CoT supervision can also affect response length and its effectiveness: prior work reports learning difficulties and excessively long reasoning under Long-CoT distillation, while longer reasoning traces can accompany error accumulation and performance degradation in smaller models (Yin et al., 2025; Luo et al., 2025). SFT can also change the diversity of sampled reasoning traces, with declining diversity accompanying higher pass@1 but lower pass@k (Dang et al., 2025).

6.3 RLVR Reasoning Boundary

RLVR can improve pass@1 while large-budget sampling recovers correct Base-model solutions (Yue et al., 2025); CoT-Pass@K studies also report expanded reasoning boundaries (Wen et al., 2025). Large-scale RL training can also produce broader reflection, retry, and self-checking behavior (Guo et al., 2025). Our Section 4 measure is deliberately narrower: it counts only markers that directly verify an intermediate or final result, so trends in our strict verification metric need not match trends in broader reflection or retry language. Other studies examine self-improvement limits (Song et al., 2025), Base-model and optimization effects (Liu et al., 2025), and amplification of pretrained behaviors (Zhao et al., 2025). RL can change strategies and knowledge use (Wang et al., 2025), while process rewards can guide exploration (Setlur et al., 2024). Shortcut learning and goal misgeneralization concern failures under distribution shift (Geirhos et al., 2020; Langosco et al., 2023); reward hacking and overoptimization concern divergence between proxy rewards and intended performance (Skalse et al., 2022; Gao et al., 2022).

7 Discussion and Limitations

Matching training-problem order and optimizer updates controls data exposure without equalizing effective optimization: SFT and GRPO use different learning signals and schedules. SFT's later OOD advantage persists at a lower nominal learning rate (Appendix B.3). The comparison is update-matched rather than compute-matched (Appendix E).

GRPO repeatedly traverses the same finite set of 7,500 problems over six epochs. Its highest observed OOD scores occur early (Table 2), and its accuracy on the fixed response-analysis set peaks at epoch 2. Data repetition may affect later policy behavior, but is not separately ablated, so we cannot attribute accuracy, length, or verification-marker changes to repetition alone.

Our verification measure counts language directly checking intermediate or final results, excluding generic *wait*, *rethink*, and retry expressions. Later GRPO declines concern this specific measure and configuration; they do not establish that GRPO generally suppresses broader reflection or retry behavior.

Appendix C.4 shows that Qwen3-8B matches longer supervised responses much less often, especially beyond 8,192 tokens. The useful length of Long-CoT supervision may therefore depend on model capability, although longer-target groups are also harder for the Base model.

Our twelve-epoch runs do not show GRPO overtaking SFT again, but cannot rule this out under other optimization settings. The controlled game highlights path dependence: outcome-reward optimization can amplify reward-compatible behaviors accessible to the policy. Experiments cover limited model families, one main GRPO learning rate, and one run per configuration; broader tasks, configurations, seeds, and shortcuts remain open.

8 Conclusion

Across matched multi-epoch training, GRPO is often stronger or comparable at early checkpoints, but sufficient training reveals a sustained Reasoning SFT OOD advantage across consecutive checkpoints. Comparisons restricted to early checkpoints can therefore underestimate the OOD performance SFT can attain with continued training. Further training weakens SFT's OOD performance, although GRPO does not overtake SFT within the observed horizon.

Response analysis suggests that the benefit of Long-CoT supervision lies in how reasoning is used rather than how long it becomes. Strict verification markers remain common in SFT OOD generations, and successful SFT responses are shorter and more verification-dense than failed ones. On the fixed response-analysis set, the SFT–GRPO density difference is already present at GRPO's highest-accuracy epoch, before its later weakening in our repeated-data run. A controlled game further shows that GRPO can amplify an ID-rewarded shortcut, achieving high ID accuracy while collapsing under paired Shortcut OOD.

AI USE STATEMENT

In this work, we used generative AI tools to assist with polishing our manuscript. We did not use generative AI tools to make final scientific judgments, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for language editing, LaTeX formatting, figure and table preparation, and reference organization. We reviewed all AI-assisted work. We take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

Chenxin An, Zhihui Xie, Xiaonan Li, Ming Zhong, Shansan Gong, Lei Li, Jun Zhang, Jingjing Xu, and Lingpeng Kong. Long chain-of-thought fine-tuning via understanding-to-reasoning transition. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 34518–34534, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1751. URL https://aclanthology.org/2025.emnlp-main.1751/.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *The Second Conference on Parsimony and Learning (Recent Spotlight Track)*, 2025. URL https://openreview.net/forum?id=d3E3LWmTar.

Xingyu Dang, Christina Baek, Kaiyue Wen, Zico Kolter, and Aditi Raghunathan. Weight ensembling improves reasoning in language models, 2025. URL https://arxiv.org/abs/2504.10478.

Bowen Ding, Yuhan Chen, Jiayang Lyv, Jiyao Yuan, Qi Zhu, Shuangshuang Tian, Dantong Zhu, Futing Wang, Heyuan Deng, Fei Mi, Lifeng Shang, and Tao Lin. Rethinking expert trajectory utilization in llm post-training for mathematical reasoning, 2026. URL https://arxiv.org/abs/2512.11470.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization, 2022. URL https://arxiv.org/abs/2210.10760.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL http://dx.doi.org/10.1038/s42256-020-00257-z.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean

Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers, 2023. URL https://arxiv.org/abs/2212.10071.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

Hugging Face. OpenR1-Math-220k. Hugging Face Datasets, 2025. URL https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

Feiyang Kang, Michael Kuchnik, Karthik Padthe, Marin Vlastelica, Ruoxi Jia, Carole-Jean Wu, and Newsha Ardalani. Quagmires in sft-rl post-training: When high sft scores mislead and what to use instead, 2025. URL https://arxiv.org/abs/2510.01624.

Dawid J. Kopiczko, Sagar Vaze, Tijmen Blankevoort, and Yuki M. Asano. Data repetition beats data scaling in long-cot supervised fine-tuning, 2026. URL https://arxiv.org/abs/2602.11149.

Vijay Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models, 2022. URL https://arxiv.org/abs/2205.05198.

Lauro Langosco, Jack Koch, Lee Sharkey, Jacob Pfau, Laurent Orseau, and David Krueger. Goal misgeneralization in deep reinforcement learning, 2023. URL https://arxiv.org/abs/2105.14111.

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, 2025. URL https://arxiv.org/abs/2503.20783.

Renjie Luo, Jiaxi Li, Chen Huang, and Wei Lu. Through the valley: Path to effective long cot training for small language models, 2025. URL https://arxiv.org/abs/2506.07712.

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning, 2024. URL https://arxiv.org/abs/2401.08967.

Kohsei Matsutani, Shota Takashiro, Gouki Minegishi, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. Rl squeezes, sft expands: A comparative study of reasoning llms, 2026. URL https://arxiv.org/abs/2509.21128.

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, 2023. URL https://arxiv.org/abs/2311.12022.

Qihan Ren, Peng Wang, Ruikun Cai, Shuai Shao, Dadi Guo, Yuejin Xie, Yafu Li, Quanshi Zhang, Xia Hu, Jing Shao, and Dongrui Liu. Rethinking generalization in reasoning sft: A conditional analysis on optimization, data, and model capability, 2026. URL https://arxiv.org/abs/2604.06628.

Yi Ren and Danica J. Sutherland. Learning dynamics of llm finetuning, 2025. URL https://arxiv.org/abs/2407.10490.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning, 2024. URL https://arxiv.org/abs/2410.08146.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 9460–9471. Curran Associates, Inc., 2022. doi: 10.52202/068431-0687. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/3d719fee332caa23d5038b8a90e81796-Paper-Conference.pdf.

Yuda Song, Hanlin Zhang, Carson Eisenach, Sham Kakade, Dean Foster, and Udaya Ghai. Mind the gap: Examining the self-improvement capabilities of large language models, 2025. URL https://arxiv.org/abs/2412.02674.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025. doi: 10.48550/arXiv.2512.13961. URL https://arxiv.org/abs/2512.13961.

Jiayu Wang, Yifei Ming, Zixuan Ke, Caiming Xiong, Shafiq Joty, Aws Albarghouthi, and Frederic Sala. Beyond accuracy: Dissecting mathematical reasoning for llms under reinforcement learning, 2025. URL https://arxiv.org/abs/2506.04723.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms, 2025. URL https://arxiv.org/abs/2506.14245.

Chao Xue, Yao Wang, Mengqiao Liu, Di Liang, Xingsheng Han, Peiyang Liu, Xianjie Wu, Chenyao Lu, Lei Jiang, Yu Lu, Haibo Shi, Shuang Liang, Minlong Peng, and Flora D. Salim. Why supervised fine-tuning fails to learn: A systematic study of incomplete learning in large language models, 2026. URL https://arxiv.org/abs/2604.10079.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388. URL https://arxiv.org/abs/2505.09388.

Huifeng Yin, Yu Zhao, Minghao Wu, Xuanfan Ni, Bo Zeng, Hao Wang, Tianqi Shi, Liangying Shao, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. Marco-o1 v2: Towards widening the distillation bottleneck for reasoning models, 2025. URL https://arxiv.org/abs/2503.01461.

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

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022. URL https://arxiv.org/abs/2203.14465.

Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining, 2025. URL https://arxiv.org/abs/2504.07912.

APPENDIX

A EXPERIMENTAL DETAILS

A.1 TRAINING DATA AND MODELS

The main training set contains a fixed set of 7,500 problems selected from OpenR1-Math-220k (Hugging Face, 2025). Reasoning SFT pairs each selected problem with its corresponding supervised response from Math-CoT-20k (Ren et al., 2026). Math-CoT-20k is based on problems from OpenR1-Math-220k and contains solutions generated by Qwen3-32B. GRPO uses the same 7,500 problems and the same problem order, but learns from policy-generated rollouts rather than supervised responses.

The main comparison starts from Qwen3-8B-Base, Qwen3-4B-Base, or Olmo3-7B. Each configuration uses one training run.

A.2 MAIN TRAINING CONFIGURATION

Table A.1 reports the settings shared by the main multi-epoch comparison. The two methods use the same training problems, order, global batch size, and update count. The maximum length is 16,384 tokens for both training and evaluation generation.

Configuration terms. An optimizer update is one parameter-update step; an epoch is one pass over the training-problem inventory. Global batch size counts problems across devices and gradient accumulation, rather than sampled responses. With batch size 64 and eight GRPO rollouts per problem, a full GRPO batch contains $64 \times 8 = 512$ responses. The reported six-epoch budget is $6 \times 118 = 708$ updates. Adam's β_1 and β_2 are the exponential-decay coefficients for its first- and second-moment estimates. Weight decay is the AdamW parameter-decay coefficient, and gradient clipping denotes the configured gradient-norm threshold. BF16 and FP32 denote bfloat16 and 32-bit floating-point arithmetic.

The peak learning rate is the maximum scheduled rate; the reported decay floor is 10% of that rate. The 70-update warm-up occupies approximately 10% of 708 updates. Rollout temperature rescales next-token logits before sampling; top-p is the cumulative-probability cutoff for nucleus sampling. One PPO epoch means one optimization pass over the rollout batch. The clipping coefficient is $\epsilon = 0.2$, giving the policy-ratio interval $[0.8, 1.2]$. Correctness reward and format bonus are the respective weights $w_{\text{corr}} = 1.0$ and $w_{\text{fmt}} = 0.1$ on binary correctness and strict-format indicators in Section 2.

Table A.1: Main training configuration. A dash marks a setting that does not apply to Reasoning SFT.

Shared settings

Configuration	Value
Base models	Qwen3-8B-Base; Qwen3-4B-Base; Olmo3-7B
Training problems	7,500
Optimizer	AdamW
Adam β_1, β_2	$(0.9, 0.999)$
Weight decay	0.01
Global batch size	64
Gradient clipping	1.0
Warm-up	70 updates (approximately 10%)
Learning-rate schedule	cosine decay to 10% of the peak rate
Epochs	6
Updates per epoch	118
Total updates	708
Maximum length	16,384
Precision	BF16 forward/backward; FP32 optimizer and reductions

Method-specific settings

Configuration	Reasoning SFT	GRPO
Peak learning rate	5×10^{-5}	1×10^{-6}
Microbatch size per GPU	1	1
Rollouts per problem	–	8
Rollout temperature	–	0.8
Top-p	–	0.95
PPO epochs per update	–	1
Clipping coefficient	–	0.2 (ratio range $[0.8, 1.2]$)
Framework loss mode	`token-mean`	`token-mean`
Correctness reward	–	1.0
Strict-format bonus	–	0.1
Required format	–	`<think>...</think><answer>...</answer>`

Microbatch loss reduction. Both methods use `token-mean` with one response per GPU microbatch. Each reduction averages that response's valid tokens, giving the $1/|y^S|$ and $1/|z_i|$ factors in Section 2. There is no pooled global-token denominator across the global batch.

A.3 Controlled State-Transition Configuration

All state-transition runs start from Qwen3-8B-Base and use 128 training problems, a global batch size of 16, a learning rate of 10^{-5}, and 64 updates. Reasoning SFT receives complete A-format state trajectories. GRPO samples eight rollouts per problem and gives the same final-answer reward to correct A- and B-format responses. This reward design makes the full transition procedure and the copy strategy equally rewarded on Shortcut ID. Table A.2 gives the remaining settings. Here one pass contains $128/16 = 8$ updates, so 64 updates give eight passes; a full GRPO batch contains $16 \times 8 = 128$ sampled responses.

Dataset construction. The game uses four states $\{0, 1, 2, 3\}$, start state 0, four operation symbols $\{\texttt{vaz}, \texttt{zob}, \texttt{khu}, \texttt{quy}\}$, and the four transition tables in Table A.3. Each vector lists the next state from current states $[0, 1, 2, 3]$. We enumerate operation sequences of length 8 and 10, execute each sequence to obtain the complete state trace and final state, and partition the resulting problem specifications by transition table, sequence length, and final state. There are $4 \times 2 \times 4 = 32$ such cells. Within each cell, problems are ordered by the SHA-256 hash of their canonical representation and assigned to non-overlapping training and evaluation roles. The reported runs use 128 unique training problems. Each 16-problem batch contains one problem for every transition-table–final-state combination, successive batches alternate between lengths 8 and 10, and the same inventory is cycled for eight passes. Reasoning SFT and GRPO receive the same problems in the same order.

Table A.2: Controlled state-transition configuration.

Configuration	Value
Base model	Qwen3-8B-Base
Training problems	128
Global batch size	16
Learning rate	1×10^{-5}
Training updates	64
Passes over the training set	8
GRPO rollouts per problem	8
Core evaluated checkpoints	0, 1, 2, 4, 8, 16, 32, 64
Additional diagnostic checkpoints	7, 60
Maximum generation length	16,384

Shortcut-ID training assigns the code for the correct final state, whereas the no-shortcut condition sets the correctness rate of the K suffix to 0%.

Table A.3: The four transition tables used to construct the controlled game. Vector entries give next states for current states $[0, 1, 2, 3]$.

Table	`vaz`	`zob`	`khu`	`quy`
C	$[1, 0, 2, 3]$	$[2, 0, 3, 1]$	$[0, 3, 1, 2]$	$[1, 0, 3, 2]$
F	$[2, 1, 3, 0]$	$[3, 2, 0, 1]$	$[3, 1, 2, 0]$	$[2, 3, 1, 0]$
L	$[1, 2, 3, 0]$	$[2, 0, 1, 3]$	$[1, 2, 0, 3]$	$[3, 2, 1, 0]$
V	$[1, 3, 0, 2]$	$[3, 0, 1, 2]$	$[2, 3, 0, 1]$	$[0, 1, 3, 2]$

Paired Shortcut ID and Shortcut OOD evaluation. The held-out evaluation set contains 128 problem specifications that do not occur in training. It is balanced across the four transition tables and four final states, with 64 sequences of each length. We use the fixed codebook $0 \mapsto$ `zwy`, $1 \mapsto$ `wyb`, $2 \mapsto$ `urz`, and $3 \mapsto$ `ksz`. For Shortcut ID, the suffix K encodes the correct final state. For each paired Shortcut OOD prompt, the suffix instead encodes $(s_L + 1) \bmod 4$. The transition table, start state, operation sequence, complete state trace, final state, and correct response are identical within each pair; only K changes. Table A.4 gives one held-out pair. The function κ maps a displayed suffix code back to its state; s_L is the state after all L operations. The expression $(s_L + 1) \bmod 4$ increments that state cyclically within $\{0, 1, 2, 3\}$. In each transition-table vector, entry $s + 1$ specifies $T(o, s)$ because states are indexed from zero.

Table A.4: A held-out Shortcut-ID/Shortcut-OOD pair. The two prompts differ only in the terminal K code. The copy response is correct on Shortcut ID and becomes incorrect on Shortcut OOD, while the complete transition response remains correct.

Field	Shortcut ID	Shortcut OOD
Transition table	`vaz` : $[1, 2, 3, 0]$; `zob` : $[2, 0, 1, 3]$; `khu` : $[1, 2, 0, 3]$; `quy` : $[3, 2, 1, 0]$	
Input sequence	`vaz zob zob khu zob khu vaz quy vaz vaz`	
K suffix	`K zwy` ($\kappa(K) = 0$)	`K wyb` ($\kappa(K) = 1$)
State trace	1, 0, 2, 0, 2, 0, 1, 2, 3, 0; final state 0	
Complete response	`A10202012302` (correct)	`A10202012302` (correct)
Copy response	`B0Z` (correct)	`B1Z` (incorrect)

No-shortcut condition. The no-shortcut condition retains the suffix K and its fixed codebook κ, but assigns a code satisfying $\kappa(K) \neq s_L$ to every training example. Thus, the direct rule $\hat{s}_L = \kappa(K)$ has 0% training accuracy and cannot earn correctness reward. This control tests whether the large Shortcut-ID gain persists when direct copying is never correct; it does not assert statistical independence between K and s_L.

A.4 Main OOD Evaluation Protocol

AIME 2024/2025 denote the corresponding annual editions of the American Invitational Mathematics Examination. At each checkpoint, we draw ten responses per problem using fixed prompts, decoding settings, and seeds across methods and checkpoints, with a maximum generation length of 16,384 tokens. Let Q be the nonempty evaluation problem set and $|Q|$ its number of problems. For problem $q \in Q$ and sample index $j \in \{1, \ldots, 10\}$, let $c_{qj} \in \{0, 1\}$ indicate final-answer correctness, with $c_q = \sum_{j=1}^{10} c_{qj}$. We compute avg10 and the pass@8 estimator of Chen et al. (2021) as

$$\text{avg10} = \frac{1}{10|Q|} \sum_{q \in Q} c_q, \qquad \text{pass@8} = \frac{1}{|Q|} \sum_{q \in Q} \left[1 - \frac{\binom{10-c_q}{8}}{\binom{10}{8}} \right],$$

Here $\binom{n}{8} = n!/[8!(n-8)!]$ counts eight-element subsets when $n \geq 8$, and is zero for $n < 8$. Thus avg10 averages correctness over all ten samples and all problems. The pass@8 term subtracts the fraction of eight-response subsets containing no correct response, then averages over problems. Multiplication by 100 converts both quantities to the percentages reported in tables; a difference between two percentages is measured in percentage points (pp). Training rollouts use the parameters in Table A.1.

B Additional Results for the Multi-Epoch SFT–GRPO Comparison

B.1 Full Numerical Results

For the Qwen models, AIME 2024 and AIME 2025 show pronounced early-to-later strengthening of Reasoning SFT. Later checkpoints also show SFT advantages on GPQA-D and MMLU-Pro. On Qwen3-4B MMLU-Pro, SFT already slightly exceeds GRPO at the representative early checkpoint, and the gap subsequently widens.

Tables B.1–B.3 report Qwen results across evaluated checkpoints and Olmo3-7B results at epochs 1–6 on AIME 2024, AIME 2025, GPQA-D, and MMLU-Pro. Dataset names form the rows, while checkpoints form the columns. Algorithm and metric are separated in the row hierarchy to avoid repeating shared labels. Base columns report the shared step-0 evaluation. Values are percentages rounded to one decimal place, and a dash denotes an unavailable evaluation. The Qwen3-8B table also includes the further-training checkpoints through step 1,416. Metrics use the denominators in Appendix A.4. A paired entry a/b elsewhere in the paper reports avg10/pass@8 at the same checkpoint; neither metric is maximized separately within that entry. A highest-observed avg10 is a maximum over the stated evaluated checkpoints, rather than an average over them.

Table B.1: Full Qwen3-8B results. Values are percentages; pass@8 is the unbiased estimate computed from all ten samples. The final panel reports further-training checkpoints through epoch 12.

Base and early checkpoints

Algorithm	Metric	Dataset	Base	1	2	4	8	16	32	64	88
Reasoning SFT	avg10	AIME 2024	13.0	14.0	12.3	9.7	8.0	16.7	17.0	17.0	15.0
		AIME 2025	9.7	10.3	8.3	11.7	12.0	12.3	19.0	18.0	17.7
		GPQA-D	37.3	38.2	35.9	35.8	34.3	29.3	39.6	39.1	32.7
		MMLU-Pro	54.7	52.1	52.5	52.1	45.2	28.0	50.4	49.4	51.1
	pass@8	AIME 2024	30.6	33.9	32.7	30.4	27.9	52.2	37.1	34.7	28.0
		AIME 2025	19.9	22.6	27.3	27.9	37.0	31.3	34.7	32.7	28.6
		GPQA-D	84.8	84.2	80.3	85.9	87.9	72.9	77.6	76.0	68.6
		MMLU-Pro	85.3	83.4	85.0	86.0	86.5	75.5	80.4	82.6	79.8
GRPO	avg10	AIME 2024	13.0	13.0	10.3	10.0	12.3	11.3	12.0	14.7	16.0
		AIME 2025	9.7	10.3	9.7	11.0	12.3	9.3	8.7	13.0	13.0
		GPQA-D	37.3	36.9	39.3	35.6	36.3	37.4	39.6	46.2	42.2
		MMLU-Pro	54.7	52.1	54.2	52.5	51.1	52.4	54.0	58.0	57.2
	pass@8	AIME 2024	30.6	30.6	27.3	26.7	28.5	24.7	24.7	35.9	37.3
		AIME 2025	19.9	22.6	27.9	30.0	24.7	32.5	27.9	25.8	28.5
		GPQA-D	84.8	80.6	85.6	81.0	80.7	83.6	85.2	86.8	83.9
		MMLU-Pro	85.3	83.4	87.8	83.1	83.6	82.7	83.9	86.3	81.8

Table B.1 continued.

Multi-epoch checkpoints

Algorithm	Metric	Dataset	118	147	176	205	236	354	472	590	708
Reasoning SFT	avg10	AIME 2024	16.3	23.7	22.7	21.3	24.3	26.7	32.7	33.0	32.7
		AIME 2025	18.0	22.3	21.3	24.0	22.0	22.0	29.3	26.3	28.0
		GPQA-D	37.8	42.5	43.3	46.3	43.0	48.9	48.6	50.1	48.8
		MMLU-Pro	50.6	60.8	57.7	62.7	60.9	64.9	64.7	66.1	66.5
	pass@8	AIME 2024	40.4	47.9	50.4	38.6	48.4	57.7	63.2	64.5	64.4
		AIME 2025	34.6	41.8	39.2	38.6	41.2	35.3	47.9	42.4	43.3
		GPQA-D	68.0	75.2	72.2	79.2	78.6	83.6	86.8	85.0	83.7
		MMLU-Pro	82.5	86.0	84.1	85.5	87.3	86.2	83.0	84.7	87.6
GRPO	avg10	AIME 2024	18.7	19.7	17.3	14.0	20.0	16.7	17.0	14.3	15.3
		AIME 2025	15.7	18.0	17.3	11.3	15.7	15.3	11.3	12.3	13.3
		GPQA-D	42.7	45.5	44.7	38.4	46.6	42.9	41.1	38.0	39.2
		MMLU-Pro	55.2	56.9	57.2	54.7	57.8	55.5	54.1	54.4	55.2
	pass@8	AIME 2024	29.2	32.6	34.5	28.0	45.9	34.6	34.6	33.3	33.9
		AIME 2025	28.6	34.4	28.0	25.8	39.8	34.5	25.9	25.3	28.6
		GPQA-D	81.1	81.1	79.7	76.5	77.1	77.4	82.3	81.4	80.2
		MMLU-Pro	84.0	81.9	81.7	84.4	80.5	78.9	80.5	82.5	81.1

Table B.1 continued.

Further-training checkpoints

Algorithm	Metric	Dataset	826	944	1062	1180	1298	1416
Reasoning SFT	avg10	AIME 2024	32.3	27.3	24.0	25.3	25.7	28.3
		AIME 2025	28.3	22.7	24.0	25.7	22.3	22.0
		GPQA-D	46.2	46.3	44.4	45.4	44.6	44.1
		MMLU-Pro	62.6	60.2	61.8	61.7	61.8	61.4
	pass@8	AIME 2024	65.0	51.3	53.7	49.9	58.5	53.9
		AIME 2025	51.2	41.3	44.0	48.4	32.0	40.0
		GPQA-D	84.6	84.8	82.8	83.7	83.9	79.2
		MMLU-Pro	86.0	87.2	86.7	84.7	88.6	85.5
GRPO	avg10	AIME 2024	12.7	11.7	13.0	12.0	11.0	13.3
		AIME 2025	13.3	11.7	11.3	9.3	12.0	10.3
		GPQA-D	39.9	36.7	37.6	33.0	35.0	36.3
		MMLU-Pro	53.8	53.3	55.2	53.6	52.9	52.8
	pass@8	AIME 2024	24.6	22.6	28.5	22.0	22.6	22.6
		AIME 2025	30.7	27.9	27.9	25.2	28.6	31.8
		GPQA-D	82.9	83.0	78.3	78.0	78.5	77.4
		MMLU-Pro	78.7	78.5	77.8	79.7	83.8	82.2

Table B.2: Full Qwen3-4B results. Values are percentages; pass@8 is the unbiased estimate computed from all ten samples.

Base and early checkpoints

Algorithm	Metric	Dataset	Base	1	2	4	8	16	32	64	88
Reasoning SFT	avg10	AIME 2024	9.7	10.3	9.3	8.0	7.7	13.7	13.7	10.7	12.3
		AIME 2025	8.7	9.7	7.3	6.0	4.0	16.7	16.7	14.0	13.0
		GPQA-D	27.4	26.7	26.5	27.3	27.9	29.4	28.8	27.3	23.4
		MMLU-Pro	36.8	36.8	35.9	34.9	33.2	45.6	50.2	51.2	50.4
	pass@8	AIME 2024	24.7	24.7	29.9	26.7	25.9	34.5	33.9	27.9	33.3
		AIME 2025	30.5	27.9	25.2	21.2	23.7	34.7	32.6	28.5	25.3
		GPQA-D	76.8	77.8	74.6	77.8	85.1	73.3	72.9	69.0	54.4
		MMLU-Pro	80.5	73.6	78.2	77.2	78.8	85.6	83.7	86.0	81.4
GRPO	avg10	AIME 2024	9.7	9.3	10.7	11.7	10.0	10.3	9.7	9.0	6.0
		AIME 2025	8.7	6.0	4.7	5.0	6.7	6.0	7.3	8.7	9.3
		GPQA-D	27.4	28.4	26.7	29.6	27.0	26.2	30.3	37.3	23.7
		MMLU-Pro	36.8	30.9	33.1	31.7	33.0	31.2	32.9	41.1	48.0
	pass@8	AIME 2024	24.7	29.3	25.3	25.9	27.3	24.6	27.9	27.3	15.3
		AIME 2025	30.5	21.3	21.8	21.2	23.3	26.7	21.9	22.6	24.7
		GPQA-D	76.8	76.0	75.1	80.6	75.6	73.9	80.8	82.9	57.8
		MMLU-Pro	80.5	74.1	78.4	75.8	78.0	75.0	74.0	75.6	76.8

Table B.2 continued.

Multi-epoch checkpoints

Algorithm	Metric	Dataset	118	147	176	205	236	354	472	590	708
Reasoning SFT	avg10	AIME 2024	14.0	20.0	18.3	17.3	18.0	24.3	25.3	24.7	27.0
		AIME 2025	16.7	17.3	20.7	17.3	17.7	19.7	22.0	22.3	21.0
		GPQA-D	28.8	36.6	34.7	36.1	38.8	42.6	44.7	44.4	44.5
		MMLU-Pro	53.8	56.2	57.7	54.8	56.3	57.5	64.4	63.2	60.7
	pass@8	AIME 2024	28.6	41.0	37.9	35.3	37.9	46.6	54.4	51.8	52.3
		AIME 2025	32.7	34.6	38.7	31.9	32.6	32.6	41.8	41.3	37.9
		GPQA-D	69.1	73.1	72.9	77.3	80.4	81.0	80.2	84.8	81.8
		MMLU-Pro	84.4	86.4	85.4	81.8	89.4	85.1	87.2	87.3	87.5
GRPO	avg10	AIME 2024	10.7	10.0	7.0	8.0	6.3	7.3	8.7	9.0	8.3
		AIME 2025	11.3	5.7	6.0	7.0	2.7	4.0	2.7	4.0	3.0
		GPQA-D	28.3	34.0	30.4	35.5	40.9	41.3	37.3	38.3	39.2
		MMLU-Pro	48.7	50.7	48.1	50.2	51.6	54.2	50.3	49.6	48.8
	pass@8	AIME 2024	23.2	25.1	21.2	18.6	15.3	24.4	19.2	18.7	21.3
		AIME 2025	28.0	27.6	22.4	25.1	16.7	12.6	11.3	16.7	14.7
		GPQA-D	71.9	75.6	69.8	77.1	77.5	76.7	66.7	70.0	75.0
		MMLU-Pro	77.8	80.5	79.6	77.9	77.3	76.6	76.7	75.1	75.5

Olmo3-7B epoch results. Table B.3 reports the shared Base evaluation and only the six epoch-end checkpoints, at steps 118, 236, 354, 472, 590, and 708. Reasoning SFT exceeds GRPO on both metrics across all four benchmarks at epochs 2–6. At epoch 1, GRPO leads on GPQA-D, while SFT is higher on AIME 2024, AIME 2025, and MMLU-Pro. At epoch 6, SFT's avg10 advantages are 22.9, 19.8, 12.3, and 22.3 percentage points, respectively. These epoch-level results extend the later SFT advantage beyond the Qwen family.

Table B.3: Olmo3-7B results at Base and epoch-end checkpoints only. E1–E6 correspond to steps 118–708 in increments of 118. Scores are percentages; pass@8 is the unbiased estimate from ten samples. Bold identifies the higher method score for each metric and epoch; the step-0 Base evaluation is shared.

Dataset	Metric	Method	Base 0	E1 118	E2 236	E3 354	E4 472	E5 590	E6 708
AIME 2024	avg10	Reasoning SFT	1.7	**22.0**	**27.7**	**23.7**	**31.3**	**35.0**	**33.3**
		GRPO	1.7	12.0	14.6	11.8	12.5	9.7	10.4
	pass@8	Reasoning SFT	13.3	**50.5**	**57.9**	**50.4**	**70.8**	**68.3**	**69.3**
		GRPO	13.3	26.0	35.0	29.5	28.5	25.0	27.5
AIME 2025	avg10	Reasoning SFT	1.0	**20.7**	**19.7**	**20.7**	**23.7**	**28.0**	**29.0**
		GRPO	1.0	10.5	12.7	10.2	10.9	8.5	9.2
	pass@8	Reasoning SFT	8.0	**35.2**	**35.2**	**38.6**	**36.5**	**42.0**	**45.3**
		GRPO	8.0	26.5	31.5	26.0	24.5	20.0	23.5
GPQA-D	avg10	Reasoning SFT	12.9	20.6	**33.0**	**31.6**	**33.4**	**36.9**	**39.3**
		GRPO	12.9	**27.0**	30.4	27.8	29.0	25.8	27.0
	pass@8	Reasoning SFT	55.9	59.7	**79.5**	**80.7**	**78.0**	**81.8**	**83.7**
		GRPO	55.9	**74.0**	76.0	73.0	74.5	70.5	72.0
MMLU-Pro	avg10	Reasoning SFT	6.8	**54.7**	**61.4**	**57.5**	**64.3**	**61.1**	**61.0**
		GRPO	6.8	41.0	43.8	40.0	41.3	37.8	38.7
	pass@8	Reasoning SFT	23.9	**82.3**	**84.8**	**79.8**	**89.2**	**81.9**	**84.6**
		GRPO	23.9	71.5	75.0	70.5	72.0	67.5	69.0

B.2 Expanded OOD Training Trajectories

The following figures expand the Qwen panels of Figure 1 and additionally report the Olmo3-7B trajectories. Each model–benchmark pair appears as a separate full-width display so that the closely spaced early checkpoints remain legible. The Qwen panels space evaluated checkpoints equally while retaining actual optimizer-step labels. Olmo3-7B panels show the shared Base evaluation followed by epoch checkpoints E1–E6 (steps 118–708). Method and metric encodings are identified in each legend.

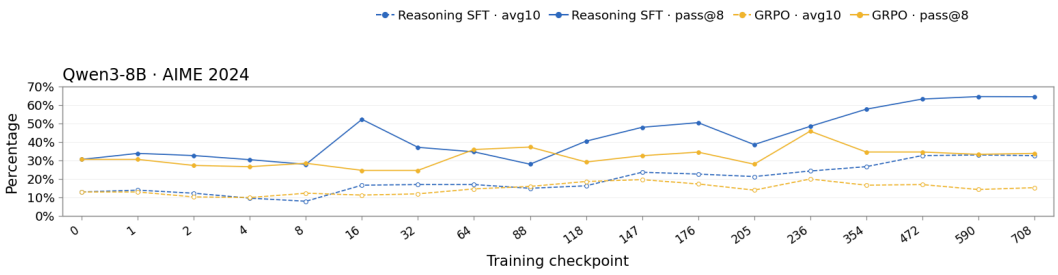


Figure B.1: Expanded Qwen3-8B trajectory on AIME 2024 through step 708. Unavailable evaluations are left blank.

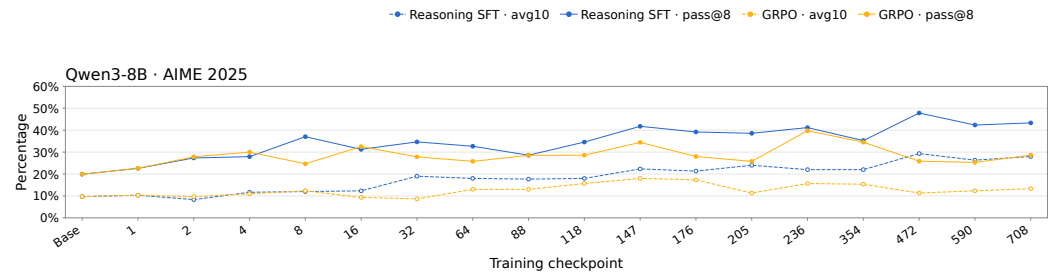


Figure B.2: Expanded Qwen3-8B trajectory on AIME 2025 through step 708. Unavailable evaluations are left blank.

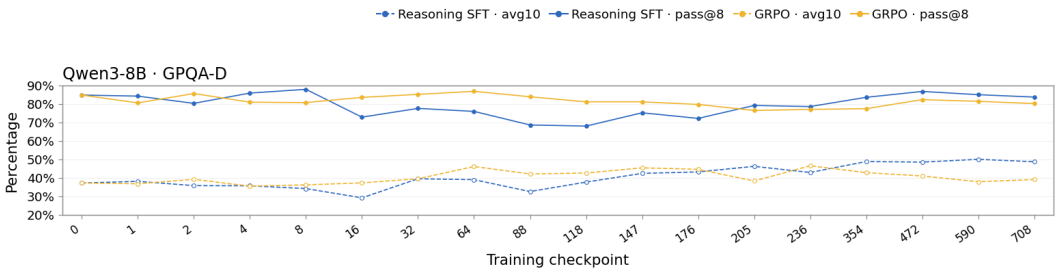


Figure B.3: Expanded Qwen3-8B trajectory on GPQA-D through step 708. Unavailable evaluations are left blank.

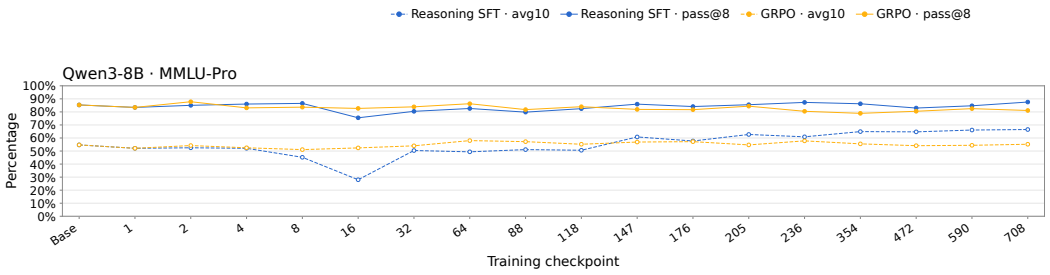


Figure B.4: Expanded Qwen3-8B trajectory on MMLU-Pro through step 708. Unavailable evaluations are left blank.

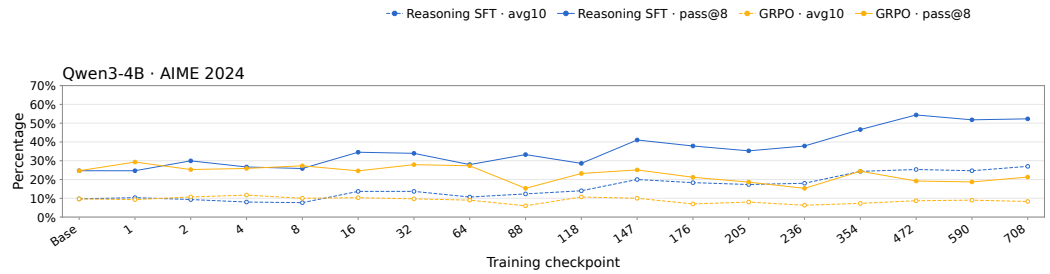


Figure B.5: Expanded Qwen3-4B trajectory on AIME 2024 through step 708. Unavailable evaluations are left blank.

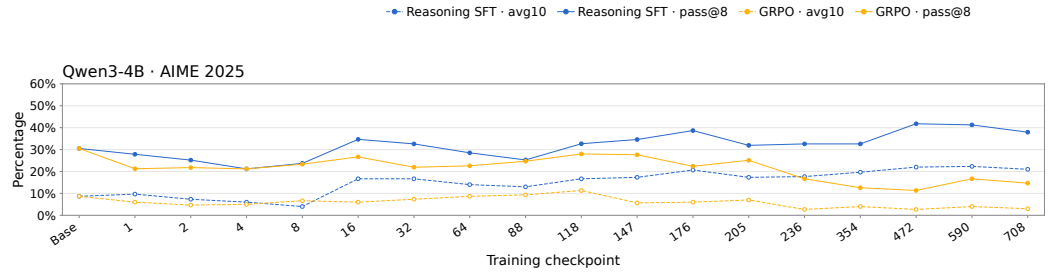


Figure B.6: Expanded Qwen3-4B trajectory on AIME 2025 through step 708. Unavailable evaluations are left blank.

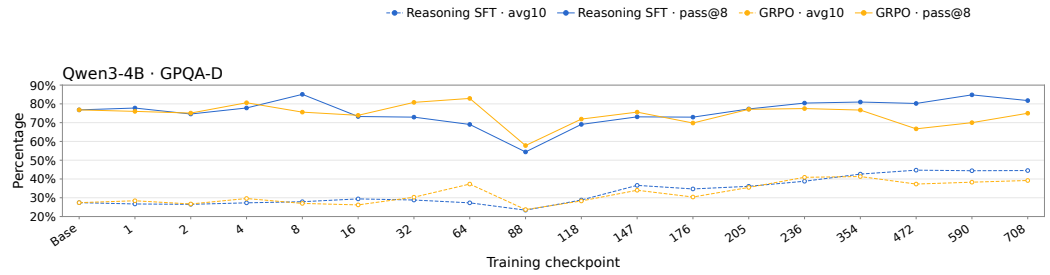


Figure B.7: Expanded Qwen3-4B trajectory on GPQA-D through step 708. Unavailable evaluations are left blank.

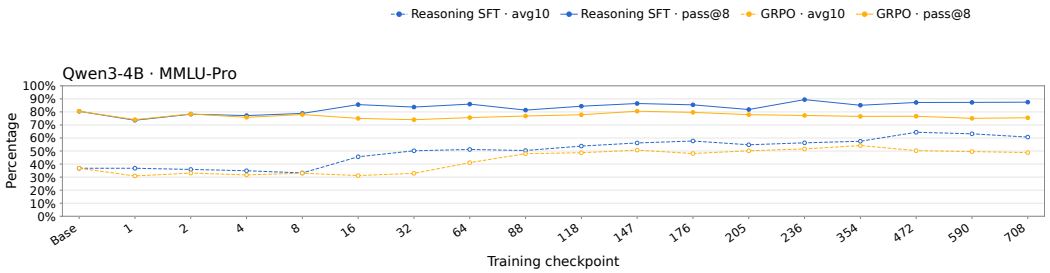


Figure B.8: Expanded Qwen3-4B trajectory on MMLU-Pro through step 708. Unavailable evaluations are left blank.

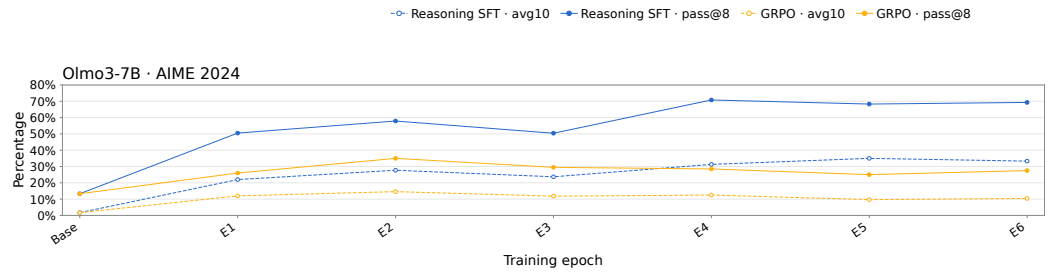


Figure B.9: Olmo3-7B results on AIME 2024 at Base and epoch checkpoints E1–E6 (steps 118–708); both methods share the step-0 evaluation.

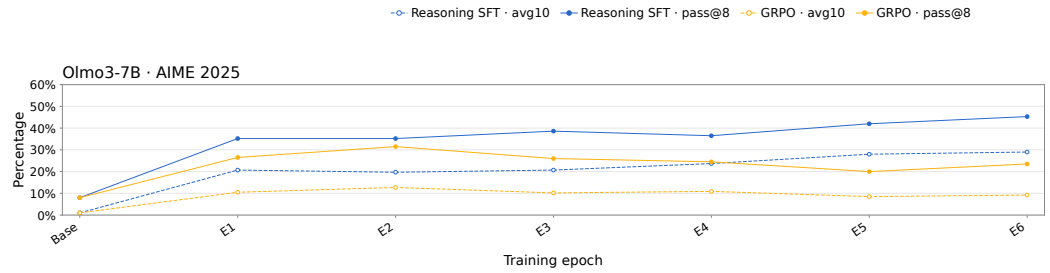


Figure B.10: Olmo3-7B results on AIME 2025 at Base and epoch checkpoints E1–E6 (steps 118–708); both methods share the step-0 evaluation.

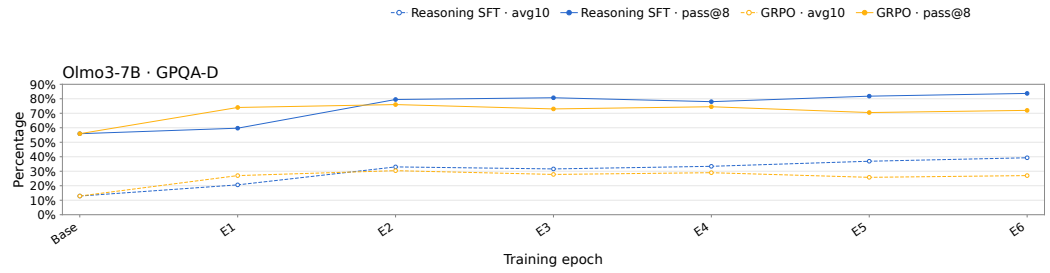


Figure B.11: Olmo3-7B results on GPQA-D at Base and epoch checkpoints E1–E6 (steps 118–708); both methods share the step-0 evaluation.

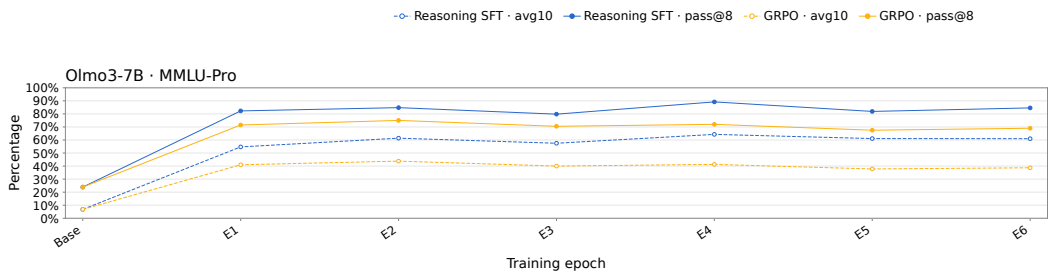


Figure B.12: Olmo3-7B results on MMLU-Pro at Base and epoch checkpoints E1–E6 (steps 118–708); both methods share the step-0 evaluation.

B.2.1 Further-Training Checkpoints

The Qwen3-8B further-training figures begin at step 236 and continue through step 1,416. One epoch contains 118 updates. The vertical line marks step 708, the end of the primary six-epoch comparison. Reasoning SFT and GRPO are evaluated at the same six further-training checkpoints.

The further-training decline is not monotonic. AIME 2025 improves further at step 826, while on MMLU-Pro avg10 and pass@8 peak at different later checkpoints. Thus, the weakening reported in Section 3.3 describes the longer-horizon trend rather than a monotonic decrease immediately after step 708. Table B.1 reports avg10 and pass@8 at steps 708 and 1,416 for all four benchmarks. To express a decline as positive, let $A_b(t)$ be the Reasoning SFT avg10 percentage before rounding on benchmark b at optimizer step t and define $D_b = A_b(708) - A_b(1416)$. Rounded to one decimal, the declines are $D_b = 4.3, 6.0, 4.7, 5.1$ pp in benchmark order AIME 2024, AIME 2025, GPQA-D, and MMLU-Pro; these are absolute percentage-point changes, not relative percentage reductions.

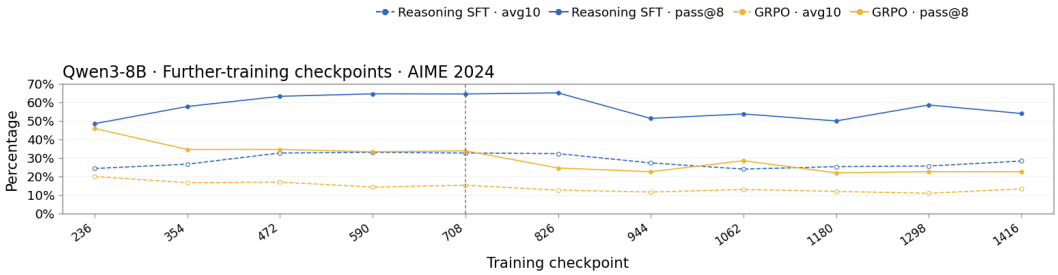


Figure B.13: Qwen3-8B further-training trajectories on AIME 2024.

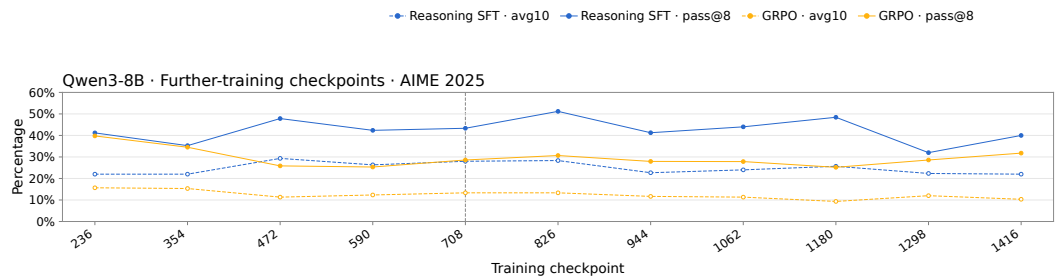


Figure B.14: Qwen3-8B further-training trajectories on AIME 2025.

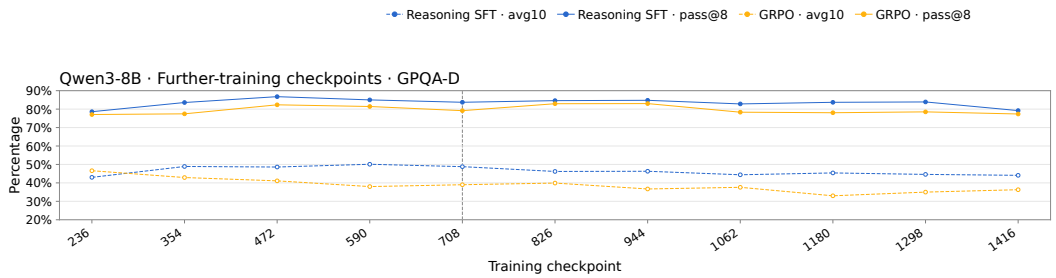


Figure B.15: Qwen3-8B further-training trajectories on GPQA-D.

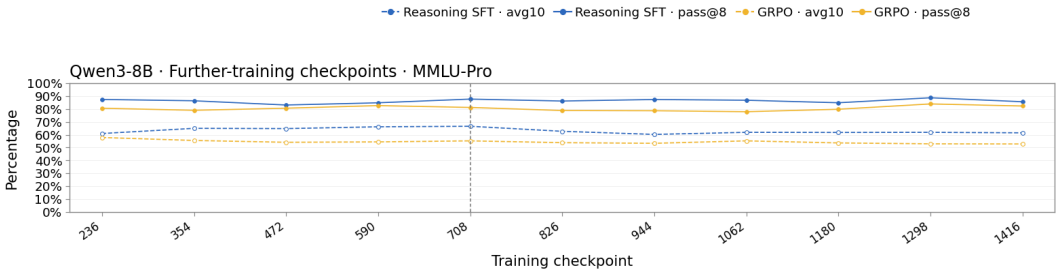


Figure B.16: Qwen3-8B further-training trajectories on MMLU-Pro.

B.3 SFT Learning-Rate Ablation

We change only the Reasoning SFT peak learning rate from 5×10^{-5} to 10^{-6} in the full Qwen3-8B protocol. Both runs use the same 7,500 training problems, 708 updates, and evaluation procedure. The main GRPO trajectory is included as a reference. All evaluation generations use a maximum length of 16,384 tokens.

Figure B.17 reports AIME 2024 and MMLU-Pro. At step 708, the 10^{-6} Reasoning SFT run reaches 25.0% avg10 and 58.4% pass@8 on AIME 2024, and 64.6%/86.3% on MMLU-Pro. The corresponding 5×10^{-5} results are 32.7%/64.4% and 66.5%/87.6%. The lower learning rate preserves the recovery during multi-epoch training, while producing a smaller improvement within the same update budget.

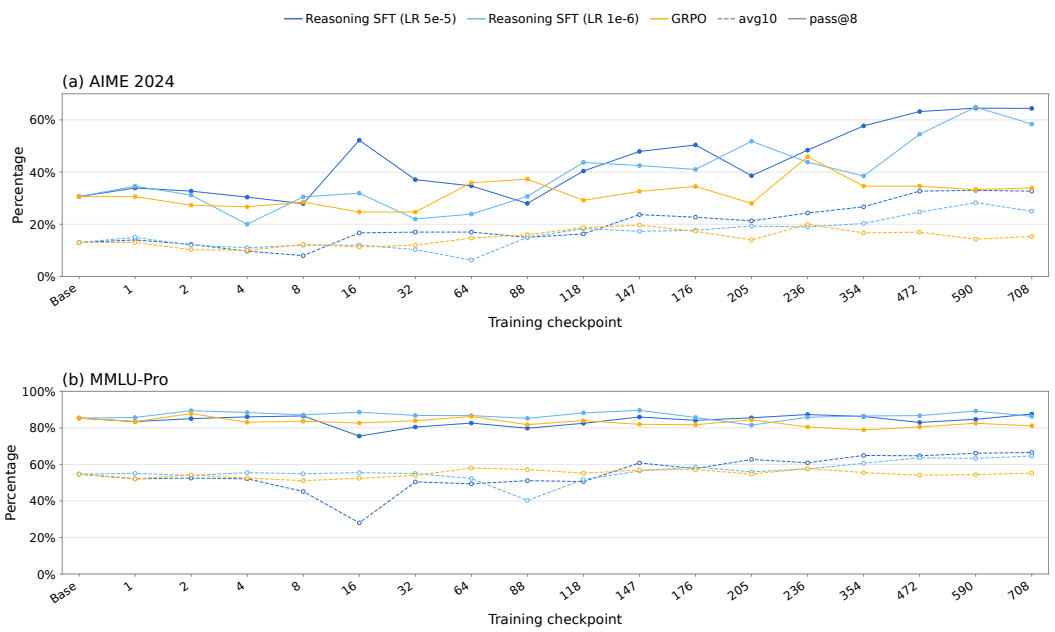


Figure B.17: Reasoning SFT learning-rate ablation on Qwen3-8B. Colors identify the two Reasoning SFT learning rates and the main GRPO reference; dashed and solid lines show avg10 and unbiased pass@8, respectively. Every completed evaluation is plotted, and missing evaluations are left blank.

Table B.4 reports the exact values at every evaluated checkpoint for the 10^{-6} run. The corresponding 5×10^{-5} Reasoning SFT and GRPO trajectories are already reported in Table B.1 and are not duplicated here.

Table B.4: Full numerical results for Qwen3-8B Reasoning SFT with a peak learning rate of 10^{-6}. Base denotes step 0. Values are percentages; pass@8 is the unbiased estimate computed from all ten samples. The corresponding main Reasoning SFT (5×10^{-5}) and GRPO results are reported in Table B.1.

Base and early checkpoints

Algorithm	Metric	Dataset	Base	1	2	4	8	16	32	64	88
Reasoning SFT	avg10	AIME 2024	13.0	15.0	12.0	11.0	12.0	12.0	10.3	6.3	15.0
		MMLU-Pro	54.7	55.1	53.9	55.5	54.9	55.5	55.0	52.3	40.3
	pass@8	AIME 2024	30.6	34.6	31.2	20.0	30.5	31.9	22.0	23.9	30.7
		MMLU-Pro	85.3	85.7	89.4	88.4	87.1	88.6	86.8	86.7	85.2

Multi-epoch checkpoints

Algorithm	Metric	Dataset	118	147	176	205	236	354	472	590	708
Reasoning SFT	avg10	AIME 2024	18.3	17.3	17.7	19.3	19.0	20.3	24.7	28.3	25.0
		MMLU-Pro	51.7	56.4	58.6	55.9	57.5	60.6	63.6	63.3	64.6
	pass@8	AIME 2024	43.7	42.5	41.0	51.8	43.8	38.5	54.5	64.9	58.4
		MMLU-Pro	88.2	89.6	85.7	81.5	85.8	86.5	86.7	89.2	86.3

C Response-Level Analyses of SFT and GRPO

This appendix provides the definitions and full numerical results for the response-level analyses in Section 4. We first examine verification behavior on OOD benchmarks and the verification-rate reweighting used in the main text. We then compare SFT and GRPO response length, verification density, termination, and correct-versus-incorrect behavior across matched training checkpoints. Additional analyses report full-response matching and performance stratified by supervised-response length.

C.1 Analysis Setup and Definitions

Sampling protocol. The matched Qwen3-8B training analysis uses 64 fixed problems and nine shared sampling seeds per problem, giving 576 responses per method at each epoch. Epochs 1–6 correspond to steps 118, 236, 354, 472, 590, and 708. The problem set, samples, and final-answer checker are fixed across comparisons. OOD analyses use stored step-708 responses on AIME 2024, AIME 2025, GPQA-D, and MMLU-Pro; evaluation settings are described in Appendix A.4. The supervised-target analysis uses fixed groups defined by target length, not generated-response length.

Strict verification markers. We use a deliberately strict set of lexical markers for language that directly checks or validates an intermediate or final result. The marker set covers verification, confirmation, and validation; double-, cross-, and sanity-checking; substituting results back into the problem; testing candidate values, solutions, or answers; boundary and special-case checks; dimensional and unit checks; and numerical checks. Generic reflection or retry expressions such as *wait*, *rethink*, and *try again*, as well as the standalone word *check*, are excluded. Markers are matched case-insensitively across the complete response. This definition measures strict verification language, not whether the checking step itself is correct.

For response r, let $h_V(r)$ be the number of verification-marker occurrences and $\ell(r) > 0$ its generated-token length. We define

$$V(r) = \mathbf{1}[h_V(r) > 0], \qquad D(r) = \frac{1{,}000 h_V(r)}{\ell(r)}.$$

Verification rate is the fraction of responses with $V(r) = 1$. **Verification density** is the mean of $D(r)$ across responses and is measured in verification-marker occurrences per 1,000 generated response tokens. Density is computed separately for each response before averaging.

Response length and termination. Response length excludes prompt tokens. We read termination directly from stored generation metadata. A response has a **normal stop** when `finish_reason=stop`. A response is **length-limited** when `finish_reason` is `length` or `max_tokens`; the generation limit is 16,384 tokens. Correctness and termination are separate properties: a normally stopped response may still contain an incorrect final answer.

Comparisons and confidence intervals. For correct-versus-incorrect comparisons, a problem is included for a method when both correct and incorrect generations are observed. For a response statistic f, we first compute the correct-minus-incorrect difference within each problem and then average these problem-level differences equally. Between-method comparisons pair the same problems and sampling seeds. Confidence intervals use problem-level bootstrap resampling, retaining a problem's sampled responses together; the matched analysis uses 5,000 resamples and the 2.5th and 97.5th percentiles. Differences are computed before rounding and need not equal differences between displayed rounded means.

Verification-rate reweighting. For OOD benchmark b, let $A_b^{S,V}$ and $A_b^{S,\neg V}$ denote SFT accuracy with and without verification, and let p_b^G denote the GRPO verification rate. We define **reweighted SFT accuracy** as

$$\widetilde{A}_b^S = p_b^G A_b^{S,V} + (1 - p_b^G) A_b^{S,\neg V}.$$

Thus, reweighting changes only how often verification occurs; SFT accuracy with and without verification remains fixed. This descriptive reweighting does not establish a causal effect of verification. The fraction of the observed gap removed is

$$R_b = \frac{A_b^S - \widetilde{A}_b^S}{A_b^S - A_b^G},$$

where A_b^S and A_b^G are observed accuracies. The mean-row value applies this ratio to the four benchmark means, rather than averaging the four ratios.

Repeated incorrect final answers. The **wrong-answer repetition rate** for a problem is the fraction of its parsed incorrect outputs equal to its most frequent incorrect final answer. We average equally over problems with an incorrect parsed answer appearing at least twice. This measures repetition of the final answer, not identity of the full reasoning response.

C.2 Verification on OOD Benchmarks

At least one strict verification marker appears in 92.24% of the Long-CoT responses used for SFT. Table C.1 examines whether this behavior remains present and associated with accuracy on OOD evaluation.

Table C.1: Strict verification on OOD benchmarks. Panels A and B use the matched step-708 checkpoint. Panel A reports verification rates and SFT accuracy with and without verification. Panel B replaces only SFT's verification rate with GRPO's corresponding rate. A value above 100% occurs when reweighted SFT falls below observed GRPO on that benchmark.

A. Verification rate and SFT conditional accuracy

Benchmark	SFT verification	GRPO verification	SFT accuracy with verification	SFT accuracy without verification	Difference (pp)
AIME 2024	73.75%	9.3%	40.40%	12.70%	+27.70 pp
AIME 2025	81.00%	7.0%	32.51%	8.77%	+23.74 pp
GPQA-D	60.69%	4.1%	52.52%	46.26%	+6.26 pp
MMLU-Pro	49.20%	5.5%	72.76%	60.43%	+12.33 pp

B. Verification-rate reweighting

Benchmark	Observed SFT	Reweighted SFT	Observed GRPO	Observed gap removed
AIME 2024	33.13%	15.28%	14.33%	94.9%
AIME 2025	28.00%	10.43%	13.33%	119.8%
GPQA-D	50.06%	46.52%	39.00%	32.0%
MMLU-Pro	66.50%	61.11%	55.20%	47.7%
Mean	**44.42%**	**33.34%**	**30.47%**	**79.4%**

C.3 Training Dynamics and Failure Patterns

Table C.2 reports matched SFT and GRPO response dynamics. Relative to epoch 1, later SFT responses are shorter and less frequently length-limited while accuracy improves. From epoch 2 onward, SFT has higher verification density than GRPO. On this fixed analysis set, GRPO accuracy is highest at epoch 2 and weakens thereafter; its responses also shorten and strict verification markers become less frequent at later checkpoints.

Table C.2: Matched SFT and GRPO response dynamics across training. Panel A reports accuracy and generated response length. Panel B reports verification rate, verification density, and the fraction of length-limited outputs under the 16,384-token limit. Brackets give 95% problem-bootstrap intervals for paired SFT-minus-GRPO differences. Density is measured per 1,000 generated tokens.

A. Accuracy and response length

Epoch	SFT accuracy	GRPO accuracy	Accuracy difference (pp)	SFT length (tokens)	GRPO length (tokens)	Length difference (tokens)
E1	66.0%	49.8%	+16.1 [8.9, 23.6]	9,556	1,511	+8,045 [6,985, 9,087]
E2†	74.1%	58.7%	+15.5 [8.2, 23.1]	8,724	1,231	+7,493 [6,428, 8,540]
E3	81.6%	58.3%	+23.3 [15.6, 30.7]	7,091	1,002	+6,089 [5,110, 7,133]
E4	84.0%	56.8%	+27.3 [20.5, 34.2]	7,456	976	+6,481 [5,549, 7,457]
E5	87.5%	54.0%	+33.5 [25.3, 42.0]	7,695	884	+6,812 [5,815, 7,820]
E6	88.4%	54.3%	+34.0 [26.6, 41.8]	7,358	911	+6,447 [5,444, 7,495]

B. Verification and length-limited generation

Epoch	Verification rate SFT / GRPO	Verification density SFT / GRPO	Density difference SFT − GRPO	Length-limited rate SFT / GRPO
E1	70.7% / 34.9%	0.376 / 0.377	−0.001 [−0.100, 0.100]	35.4% / 0%
E2†	86.5% / 21.9%	0.522 / 0.254	+0.268 [0.138, 0.408]	24.5% / 0%
E3	86.8% / 20.7%	0.597 / 0.258	+0.339 [0.207, 0.481]	12.7% / 0%
E4	81.8% / 17.7%	0.479 / 0.210	+0.269 [0.158, 0.388]	10.1% / 0%
E5	90.6% / 15.6%	0.606 / 0.207	+0.399 [0.260, 0.545]	12.5% / 0%
E6	89.9% / 16.3%	0.593 / 0.209	+0.384 [0.257, 0.518]	11.1% / 0%

†Highest GRPO accuracy on the fixed response-analysis set.

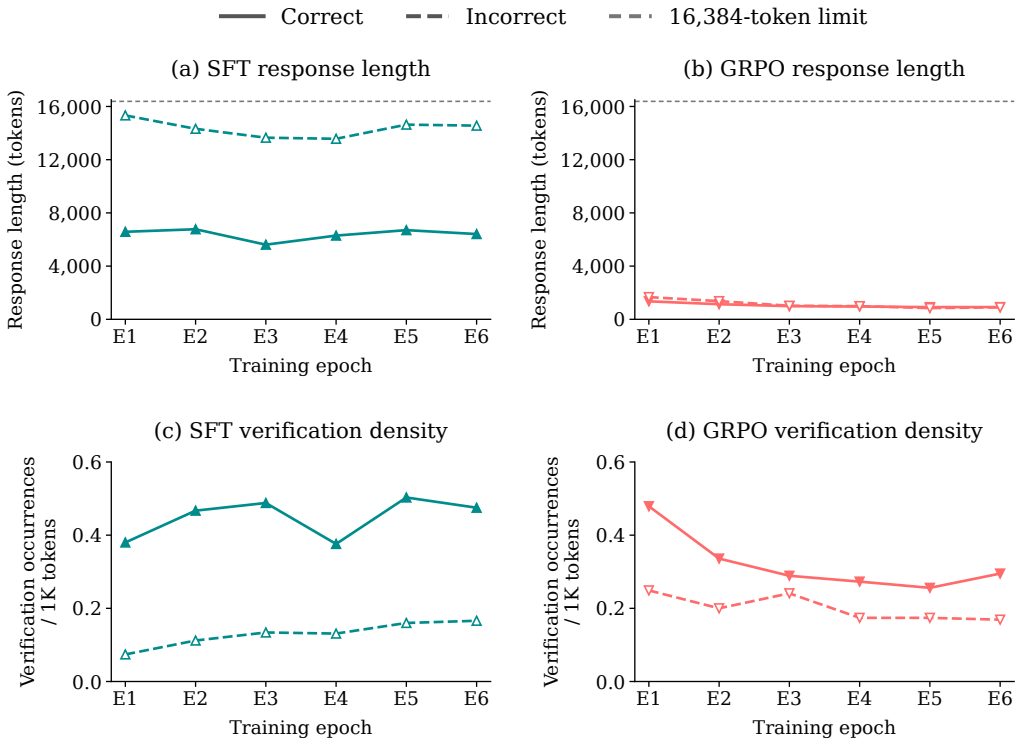


Figure C.1: Correct and incorrect responses across epochs 1–6. Top: response length for SFT and GRPO on the same 0–16,500-token scale; the dashed horizontal line marks the 16,384-token generation limit. Bottom: verification density on a shared scale, measured in verification-marker occurrences per 1,000 generated tokens. Solid lines denote correct responses and dashed lines incorrect responses. Successful SFT responses are shorter and more verification-dense than failed responses; GRPO shows much smaller length separation.

Figure C.1 separates correct and incorrect generations. Table C.3 gives SFT differences with confidence intervals, GRPO wrong-answer repetition, and an OOD comparison of SFT length behavior. The AIME 2025 length interval includes zero; its length-limit-rate interval remains negative.

Table C.3: Failure summary. Panel A reports SFT correct-minus-incorrect differences across training; Panel B reports GRPO wrong-answer repetition; Panel C reports SFT correct-minus-incorrect differences on OOD benchmarks. Brackets are 95% problem-bootstrap intervals. Wrong-answer repetition rate measures repetition of the parsed incorrect final answer, not identity of the full reasoning response.

A. SFT correct minus incorrect

Epoch	Length difference (tokens)	Verification-density difference	Length-limit-rate difference (pp)
E1	−8,749 [−9,758, −7,702]	+0.307 [0.239, 0.381]	−78.6 [−85.2, −70.5]
E2	−7,550 [−8,932, −5,982]	+0.356 [0.279, 0.441]	−67.5 [−78.9, −54.1]
E3	−8,041 [−9,157, −6,864]	+0.354 [0.262, 0.461]	−52.7 [−67.7, −37.6]
E4	−7,277 [−8,501, −5,918]	+0.245 [0.174, 0.327]	−48.8 [−62.4, −36.2]
E5	−7,930 [−9,344, −6,371]	+0.342 [0.240, 0.449]	−68.3 [−85.9, −47.4]
E6	−8,149 [−9,289, −6,968]	+0.309 [0.221, 0.411]	−58.4 [−75.2, −44.1]

B. GRPO wrong-answer repetition

Epoch	E1	E2	E3	E4	E5	E6
Wrong-answer repetition rate	27.3%	42.9%	36.3%	33.8%	30.5%	34.1%

C. OOD check of SFT length behavior

Benchmark	Length difference (tokens)	Length-limit-rate difference (pp)
AIME 2024	−3,711 [−4,991, −2,569]	−62.07 [−75.59, −46.10]
AIME 2025	−886 [−2,960, 618]	−23.04 [−47.25, −2.47]
GPQA-D	−1,805 [−2,480, −1,166]	−18.61 [−25.67, −11.38]
MMLU-Pro	−1,409 [−1,878, −952]	−9.52 [−17.16, −3.05]

C.4 Additional Response Analyses

Full-response matching. A **supervised response** is a correct generated response matching the supervised full-response pattern under the fixed matching procedure. A **Base-model response** is a correct generated response matching the Base model's full-response pattern under the same procedure. The supervised pattern uses frozen supervised examples; the Base-model pattern uses a stable cluster of correct Base generations when identifiable. Lexical features and matching rules are fixed across checkpoints and methods and are used only for post-hoc analysis. **Other-correct** denotes a correct response matching neither full-response pattern; it can include paraphrases, mixtures, or responses sharing local parts with either pattern and is not treated as a new reasoning procedure. Table C.4A reports accuracy and response-matching rates as percentages of all sampled outputs, not percentages conditional on correctness.

Supervised-target length. Table C.4B groups the fixed Qwen3-8B audit by supervised-target token length. Accuracy and E6 supervised-response rates use all generated outputs in each group; mean target length weights problems equally. SFT matches the supervised response much less often as supervised targets become longer. At epoch 6, the rate falls from 47.03% for targets up to 4,096 tokens to 1.07% for targets above 8,192 tokens. The longer-target groups also have lower Base-model accuracy.

Table C.4: Additional response analyses. Panel A reports full-response matching across training. Panel B reports accuracy and supervised-response matching by supervised-target length. All accuracy and matching rates are percentages of all outputs in the relevant group; mean target length is in tokens.

A. Full-response matching

Epoch	SFT accuracy	SFT supervised response	GRPO accuracy	GRPO Base-model response
E1	65.97%	34.72%	49.83%	28.65%
E2	74.13%	49.13%	58.68%	39.41%
E3	81.60%	55.21%	58.33%	39.76%
E4	84.03%	66.67%	56.77%	39.58%
E5	87.50%	75.52%	53.99%	37.67%
E6	88.37%	79.51%	54.34%	39.41%

B. Supervised-target length

Supervised-target tokens	Mean target	Base accuracy	E1 accuracy	E3 accuracy	E6 accuracy	E6 supervised response
$\leq$4,096	2,825	54.45%	38.75%	61.09%	68.13%	47.03%
4,097–8,192	6,019	27.46%	8.91%	19.95%	20.48%	9.84%
$>$8,192	9,900	22.33%	5.18%	11.43%	14.18%	1.07%

D State-Transition Strategy Analyses

D.1 Strategy-Level Response Dynamics

We report observable response rates on the paired Shortcut-ID and Shortcut-OOD evaluations. A full trajectory contains the complete state sequence with every transition valid under T; A-format compliance alone is insufficient. The explicit B-format copy rate is the fraction of sampled outputs matching the canonical response $\mathtt{B}\kappa(K)\mathtt{Z}$. On Shortcut ID, this agreement alone cannot distinguish following K from computing the correct answer. On paired Shortcut OOD, it counts outputs following the incorrect K, and is interpreted jointly with the intervention that changes only K. The K-following rate is the fraction of sampled outputs whose final answer agrees with $\kappa(K)$ across output formats, as reported in Appendix D.3.

Rate definitions. Each evaluation contains 128 problems and eight sampled outputs per problem, giving $N = 1{,}024$ outputs. For a response r_j to problem q_j, let F_j indicate a complete valid trajectory, B_j indicate an exact canonical B-format copy response, and K_j indicate agreement of its parsed final state with $\kappa(K_{q_j})$. A full trajectory supplies exactly L_{q_j} states and satisfies every prescribed transition from the given initial state; L_q is the number of operations in problem q. For any of these indicators I_j, its percentage in prose and tables is

$$\mathrm{Rate}(I) = \frac{100}{N} \sum_{j=1}^{N} I_j.$$

The denominator includes all sampled outputs. Figure D.1 plots the corresponding fractions, $\mathrm{Rate}(I)/100$. Accuracy uses the same formula with final-answer correctness as the indicator. A-format rate instead measures compliance with the A-output format; it does not require the transitions to be correct. Percentages of A-format, copy, full-trajectory, and K-following outputs therefore measure different, potentially overlapping properties.

Outputs matching the canonical copy response occur in 5.1% of Shortcut-ID and 4.0% of Shortcut-OOD Base samples. These observations establish that copy-compatible outputs are available under free generation; they are not used alone to establish systematic K dependence in the Base model. Their frequency increases substantially under GRPO. At checkpoint 64, the explicit B-format copy rate reaches 79.5% on Shortcut ID and 81.0% on Shortcut OOD, whereas Reasoning SFT reduces this rate to 0.0%. In contrast, Reasoning SFT increasingly produces complete and valid transition trajectories. The observed amplification is consistent with prior analyses of RL strengthening pretrained behaviors (Zhao et al., 2025); the numerical rates above are measurements from our game.

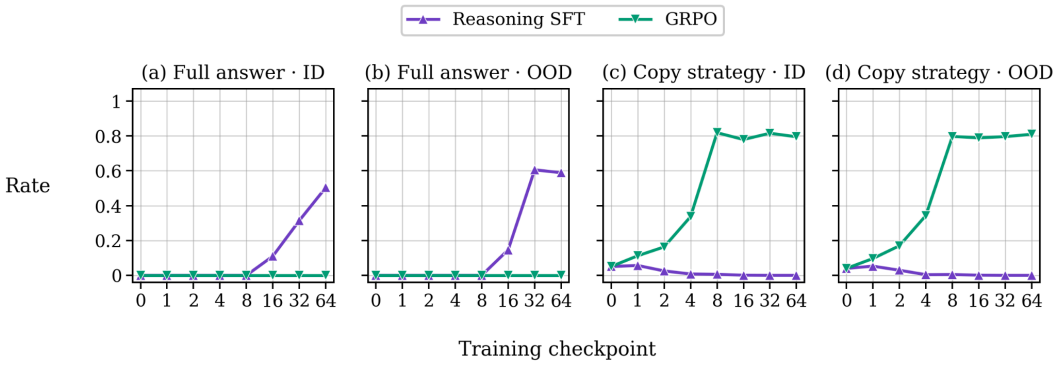


Figure D.1: Strategy-level response dynamics on paired Shortcut-ID and Shortcut-OOD evaluations. Panels (a–b) show complete, valid trajectory rates; panels (c–d) show the fraction of outputs matching the explicit B-format copy response B$\kappa(K)$Z. On Shortcut OOD, these matches follow the incorrect state encoded by K.

D.2 Shortcut Reliability and Process Rewards Do Not Prevent OOD Collapse

We test two direct ways to discourage reliance on K. The process-reward condition adds the fraction of locally valid state transitions to answer correctness. The other two conditions use answer correctness alone while making K encode the correct final state on 62.5% or 50% of training questions. All three runs start from the same Qwen3-8B-Base weights and otherwise use the state-transition configuration in Table A.2. Table D.1 summarizes the differences. For training set $\mathcal{Q}_{\mathrm{train}}$, shortcut reliability is $\alpha = |\mathcal{Q}_{\mathrm{train}}|^{-1} \sum_{q \in \mathcal{Q}_{\mathrm{train}}} \mathbf{1}[\kappa(K_q) = s_{L,q}]$, where $s_{L,q}$ is the correct final state of problem q. Thus 62.5% and 50% correspond to 80 and 64 of the 128 training questions. These are training-question fractions, not observed rollout success rates. A planned replacement at update 8 in the process-reward run was token-identical to the sampled response and therefore did not change training; we treat this run only as a process-reward condition.

These reduced-reliability training conditions are distinct from the 0% direct-copy-accuracy control in Appendix A.3. All three runs are evaluated on the same paired tests: K encodes the correct final state on Shortcut ID and an incorrect state on Shortcut OOD.

Table D.1: Additional Base-model-initialized state-transition conditions. The final column gives the fraction of training questions for which K encodes the correct final state.

Training condition	Reward	Correct K in training
Process-reward GRPO	Answer correctness + local-transition validity	100%
Outcome-only GRPO	Answer correctness	62.5%
Outcome-only GRPO	Answer correctness	50%

Figure D.2 shows the same qualitative separation in all three runs. At update 64, process-reward GRPO reaches 87.5% Shortcut-ID accuracy and 0.0% Shortcut-OOD accuracy. Outcome-only GRPO reaches 99.9%/0.0% when K is correct on 62.5% of training questions and 100.0%/0.0% when it is correct on 50%. Neither weaker shortcut reliability nor the implemented process reward prevents the final answer from becoming sensitive to K.

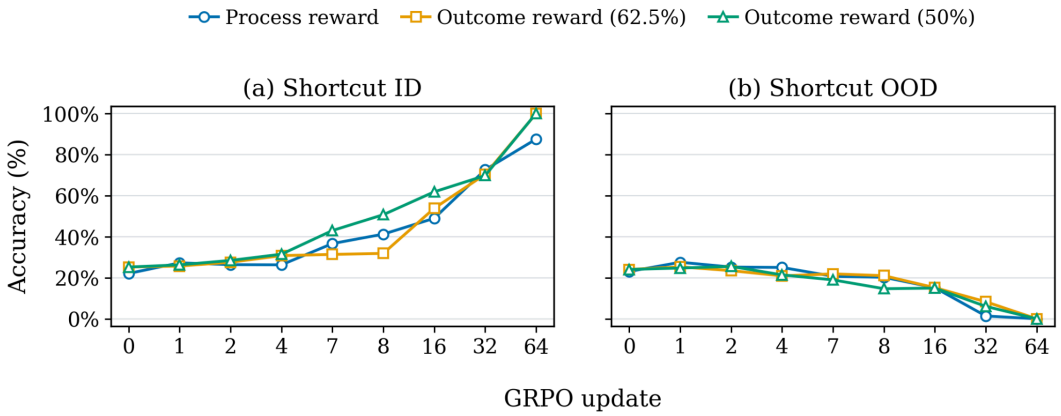


Figure D.2: Shortcut-ID and Shortcut-OOD accuracy under process reward and two outcome-only training conditions. Percentages in the legend give the fraction of training questions for which K encodes the correct final state. Horizontal positions are equally spaced evaluated GRPO updates. Each point aggregates 1,024 outputs from one fixed training run.

The 50% condition also shows why making K correct on only half of the training questions need not reverse an established dependence on K. At update 64, seven question groups contain eight correct rollouts and nine groups contain eight incorrect rollouts; no group contains both outcomes. Group normalization therefore assigns zero advantage to every rollout even though the aggregate answer reward is $100(7 \times 8)/(16 \times 8) = 43.75\%$. Each group has constant reward and hence zero within-group standard deviation; the zero-variance advantage convention is defined in Section 2. This observation explains how the behavior can persist at the endpoint. It does not explain why the model first begins to rely on K, and each condition contains only one training run.

D.3 GRPO after Four- and Sixteen-Update SFT Warm-Up

We compare GRPO initialized after four or sixteen Reasoning SFT updates on complete correct A-format trajectories. For each warm-up duration, two runs start from the same frozen checkpoint and receive 64 GRPO updates on the same Shortcut-ID training questions in the same order, where K encodes the correct final state. One run uses answer correctness alone; the other uses the additive reward $r = \mathbf{1}[\text{answer correct}] + v_{\text{local}}$, where v_{local} is the fraction of locally valid transitions. The two reward terms have coefficient one: answer correctness contributes zero or one, and local validity contributes a fraction in $[0, 1]$. For a parsed trajectory with reported states $\hat{s}_1, \ldots, \hat{s}_L$ and $\hat{s}_0 = s_0$, transition t is locally valid when $\hat{s}_t = T(o_t, \hat{s}_{t-1})$, using the preceding reported state. Complete-trajectory validity requires all prescribed transitions to be valid. All remaining settings follow Table A.2. We index downstream GRPO updates by g, with $g = 0$ denoting the corresponding SFT checkpoint. Total optimizer updates are $4 + g$ for SFT-4 and $16 + g$ for SFT-16; at $g = 64$, the runs have received 68 and 80 total updates, respectively.

Four-update warm-up: format learning followed by A-format shortcut dependence. At the SFT-4 checkpoint, 99.5% of outputs use the A-format, but no output reproduces the complete supervised trajectory. Shortcut-ID and Shortcut-OOD accuracy are 24.22% and 25.68%, respectively. Outcome-only GRPO preserves the A-format while sharply separating ID and OOD accuracy (Figure D.3, panels a–b). At $g = 64$, 99.9% of outputs use the A-format and the explicit B-format copy rate is only 0.1%, yet Shortcut-ID accuracy reaches 95.1% and Shortcut-OOD accuracy falls to 0.7%. The Shortcut-OOD K-following rate across output formats is 95.2%, measuring agreement with the incorrect state encoded by K.

With SFT-4 initialization, the local process reward delays this separation without preventing it. At $g = 16$, Shortcut-ID accuracy is about 31% and Shortcut-OOD accuracy is 24.0%, while mean local-transition validity rises from 26.2% to 34.8%. The split appears by $g = 32$, when ID accuracy reaches 76.4% and OOD accuracy falls to 8.1%. At $g = 60$, corresponding to 64 total optimizer updates, the respective accuracies are 96.2% and 0.8%, and the Shortcut-OOD K-following rate is 96.5%. At $g = 64$, ID/OOD accuracy reaches 98.2%/0.3% and the Shortcut-OOD K-following rate reaches 98.7%. Every evaluated checkpoint has a 0.0% full-trajectory rate. At $g = 64$, all outputs have the A-format, the explicit B-format copy rate is zero, and mean local-transition validity reaches 43.8%.

The SFT-4 control separates learning the response format from learning the transition policy. SFT makes the A-format readily accessible without establishing complete valid trajectories; subsequent GRPO retains that format while amplifying K-dependent answer selection. The resulting shortcut is copying within an A . . . Z response, rather than adoption of the deliberately short canonical B$\kappa(K)$Z response. Under the additive process reward, a partially valid prefix can also earn local-transition credit while the final answer follows K. Thus, the failure does not require the short B-format response introduced in the task design. These observations establish A-format shortcut dependence, not a causal preference for shorter outputs.

Sixteen-update warm-up: GRPO consolidates a policy already learned through SFT. At the SFT-16 checkpoint, before any downstream GRPO update ($g = 0$), SFT has already learned a policy that achieves 75.98% Shortcut-ID and 75.10% Shortcut-OOD accuracy (Figure D.3, panels c–d). Subsequent GRPO retains and strengthens this already successful behavior under both reward conditions, with both accuracies approaching 100% at the final checkpoints. Here, GRPO consolidates an SFT-learned policy; these results do not show that GRPO acquires the generalizing policy or repairs an OOD failure. The key contrast is SFT-4: learning the A-format alone leaves room for GRPO to reinforce an A-format shortcut, whereas the SFT-16 initialization already performs the task successfully under the paired shift. The contrast demonstrates dependence on what SFT has learned before GRPO begins, rather than a generic benefit of warm-up duration. It does not isolate the internal changes responsible for this difference, and each condition contains one training run.

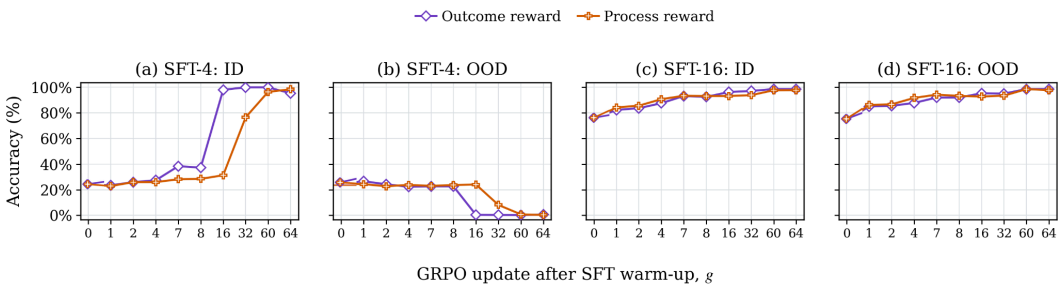


Figure D.3: Shortcut-ID and Shortcut-OOD accuracy under GRPO initialized after four or sixteen Reasoning SFT updates. Panels (a–b): SFT-4 learns the A-format, but subsequent GRPO develops A-format K dependence and OOD collapse. Panels (c–d): SFT-16 already has high ID and OOD accuracy at $g = 0$; GRPO consolidates this SFT-learned behavior rather than restoring generalization. The horizontal axis reports downstream GRPO update g, equally spaced by evaluated checkpoint. Total optimizer updates are $4 + g$ and $16 + g$, respectively. Each point aggregates 1,024 outputs from one fixed training run.

D.4 Full Numerical Results

Table D.2 reports the updated Reasoning SFT accuracy trajectories, including the Base evaluations. At update 64, SFT reaches 94.43% on both Shortcut ID and Shortcut OOD, and 94.41%/99.33% on no-shortcut ID/OOD. The SFT-4 and SFT-16 initial accuracies used at $g = 0$ in Figure D.3 match the Shortcut-ID/OOD entries at updates 4 and 16, respectively.

Table D.2: Reasoning SFT accuracy in the shortcut and no-shortcut conditions. Values are percentages; Base denotes step 0.

Evaluation	Base	1	2	4	8	16	32	64
Shortcut ID	22.95	22.95	25.39	24.22	26.56	75.98	88.09	94.43
Shortcut OOD	24.71	24.41	26.46	25.68	25.10	75.10	90.63	94.43
No-shortcut ID	23.11	23.06	25.71	23.92	25.68	77.85	82.57	94.41
No-shortcut OOD	24.79	24.66	26.05	24.13	24.45	76.97	93.51	99.33

Table D.3 reports every evaluated Shortcut-ID and Shortcut-OOD checkpoint for the three Base-model-initialized conditions and the two SFT-4 warm-up conditions. The SFT-16 accuracy trajectories are shown in Figure D.3, panels (c–d).

Table D.3: Full Shortcut-ID and Shortcut-OOD accuracy for the Base-model-initialized and SFT-4 conditions. Values are percentages. Each cell aggregates 1,024 outputs from one fixed training run. For the SFT-4 conditions, columns report downstream GRPO update g; $g = 0$ is the checkpoint after four SFT updates, $g = 60$ corresponds to 64 total optimizer updates, and $g = 64$ corresponds to 68 total optimizer updates.

Base-model initialization

Training condition	Evaluation	0	1	2	4	7	8	16	32	64
Process reward	Shortcut ID	22.1	27.2	26.4	26.3	36.6	41.2	48.9	72.6	87.5
	Shortcut OOD	23.0	27.5	25.2	25.0	20.7	20.2	15.1	1.4	0.0
Outcome reward (62.5%)	Shortcut ID	25.2	25.7	27.5	30.8	31.3	31.8	53.8	70.3	99.9
	Shortcut OOD	24.0	25.3	23.4	20.9	21.9	21.0	15.1	8.3	0.0
Outcome reward (50%)	Shortcut ID	25.2	26.3	28.4	31.4	43.0	50.7	61.8	69.7	100.0
	Shortcut OOD	24.0	24.7	25.5	21.4	18.9	14.6	14.9	6.1	0.0

Four-update SFT warm-up

Training condition	Evaluation	0	1	2	4	7	8	16	32	60	64
SFT-4 + outcome reward	Shortcut ID	24.22	23.4	25.9	27.3	38.3	37.2	97.9	99.7	99.7	95.1
	Shortcut OOD	25.68	26.6	24.2	22.3	22.6	22.6	0.4	0.3	0.2	0.7
SFT-4 + process reward	Shortcut ID	24.22	22.9	25.9	25.9	28.1	28.4	31.2	76.4	96.2	98.2
	Shortcut OOD	25.68	24.3	22.7	23.7	22.9	23.6	24.0	8.1	0.8	0.3

E TRAINING-COMPUTE ACCOUNTING AND FLOP-NORMALIZED OOD TRAJECTORIES

Reasoning SFT and GRPO use the same ordered set of 7,500 training problems and both perform 708 optimizer updates, but their training FLOPs are not matched. We distinguish three complementary accounting conventions: *valid-token FLOPs* approximate model computation on non-padding tokens; *executed-model FLOPs* additionally account for implementation-dependent computation on padded positions; and *GPU-hours* measure resource use, including system overheads not represented by a parameter–token formula. Evaluation generation, checkpoint conversion, and scoring are excluded from the training budget. We report cumulative computation in EFLOPs, with $1\ \text{EFLOP} = 10^{18}$ floating-point operations; this is an operation count, not the throughput unit EFLOP/s.

E.1 TRAINING-COMPUTE ACCOUNTING

Parameter and token counts. The "8B" in Qwen3-8B is a model-family name, not the parameter count substituted into the calculation. Summing the elements of the 399 unique weight tensors in the shared Qwen3-8B-Base checkpoint gives $P = 8{,}190{,}735{,}360$ parameters. Here P counts model weights only, excluding optimizer states, gradients, activations, and tokens. Let D denote the number of token positions processed by the model operation being counted.

Analytical FLOP convention. Counting a multiply–accumulate as two floating-point operations gives the dense forward-pass approximation $C_{\text{fwd}} \approx 2PD$. Following the convention in Hoffmann et al. (2022), the backward pass costs approximately twice the forward pass, yielding

$$C_{\text{train}} \approx (2P + 4P)D = 6PD.$$

Full activation recomputation adds approximately one forward pass during backpropagation (Korthikanti et al., 2022). For the accounting here, this gives

$$C_{\text{train,recompute}} \approx (2P + 4P + 2P)D = 8PD.$$

These are analytical approximations, not per-kernel hardware measurements. The valid-token budgets are

$$C_{\text{SFT}} = 8PD_{\text{SFT}}, \qquad C_{\text{GRPO}} \approx 2PD_{\text{rollout}} + 2PD_{\text{old}} + 8PD_{\text{update}}.$$

Here D_{SFT} counts valid SFT training tokens. For GRPO, D_{rollout} is the effective generation-token count after prefix-cache accounting, while D_{old} and D_{update} count tokens processed for actor old-log-probability recomputation and the actor forward–backward update, respectively. The generation term accounts for reused prefixes rather than charging a fresh prefix computation for every rollout.

Run totals. SFT processes 45,312 training sequences containing 279,868,086 valid tokens and 742,391,808 tensor positions after padding to a maximum length of 16,384. GRPO produces $708 \times 64 \times 8 = 362{,}496$ rollouts. Actor trajectory tokens range from 520,000 to 570,000 per update, with a central value of 545,000; over 708 updates, this is 368.16–403.56 million tokens, centrally 385.86 million. Table E.1 reports the supplied central estimates and sensitivity ranges. GPU-hours sum elapsed training hours over the participating GPUs. The ranges reflect the observed trajectory-token range and are not confidence intervals.

Table E.1: Cumulative training-compute accounting for Qwen3-8B. EFLOPs denote 10^{18} floating-point operations. Brackets give sensitivity ranges associated with the GRPO trajectory-token range, not confidence intervals. Executed-model FLOPs are implementation-aware estimates rather than hardware-counter measurements.

Accounting convention	SFT	GRPO	GRPO/SFT
Valid-token FLOPs (EFLOPs)	18.34	37.29 [35.55, 39.03]	2.03 [1.94, 2.13]
Executed-model FLOPs (EFLOPs)	47.40	37.12 [33.51, 38.85]	0.78 [0.71, 0.82]
Training GPU-hours	76.52	162.94 [154.19, 169.40]	2.13 [2.01, 2.21]

Under valid-token FLOPs and GPU-hour accounting, GRPO uses approximately twice the SFT training budget. Executed-model FLOPs give the opposite ordering because this SFT implementation

still performs dense linear operations on positions padded to 16,384. The implementation-aware estimates are reported separately from the valid-token parameter–token approximation. These conventions answer different questions and should not be collapsed into an unqualified FLOPs ratio.

E.2 FLOP-NORMALIZED OOD TRAJECTORIES

Figure E.1 plots mean sample accuracy against cumulative valid-token training compute rather than optimizer step. For an evaluation with N sampled responses and correctness indicators $c_j \in \{0, 1\}$, mean sample accuracy is $100N^{-1}\sum_{j=1}^{N} c_j$. All four panels use this measure because historical SFT evaluations used different sampling contracts, including sixteen responses per problem; we do not relabel those results as a common pass@8 estimate.

Key central compute values are 12.43 EFLOPs at GRPO step 236, 18.34 EFLOPs at SFT step 708, 18.64 EFLOPs at GRPO step 354, and 37.29 EFLOPs at GRPO step 708. GRPO step 236 is approximately compute-equivalent to SFT step 480; this is a budget correspondence, not an additional evaluated checkpoint. By GRPO step 354, cumulative compute exceeds the complete six-epoch SFT budget, so no observed SFT checkpoint has an equal or larger compute budget.

Blue circles connected by solid lines denote SFT. Orange squares belong to one GRPO run: solid segments lie within the complete SFT budget, and dashed segments continue beyond it. The line style changes at 18.34 EFLOPs, marked by the vertical gray dashed line; the gray region identifies the beyond-budget range. The style change does not denote a second experiment, a fitted curve, or extrapolation. GRPO horizontal intervals show sensitivity to the 520,000–570,000 trajectory tokens per update.

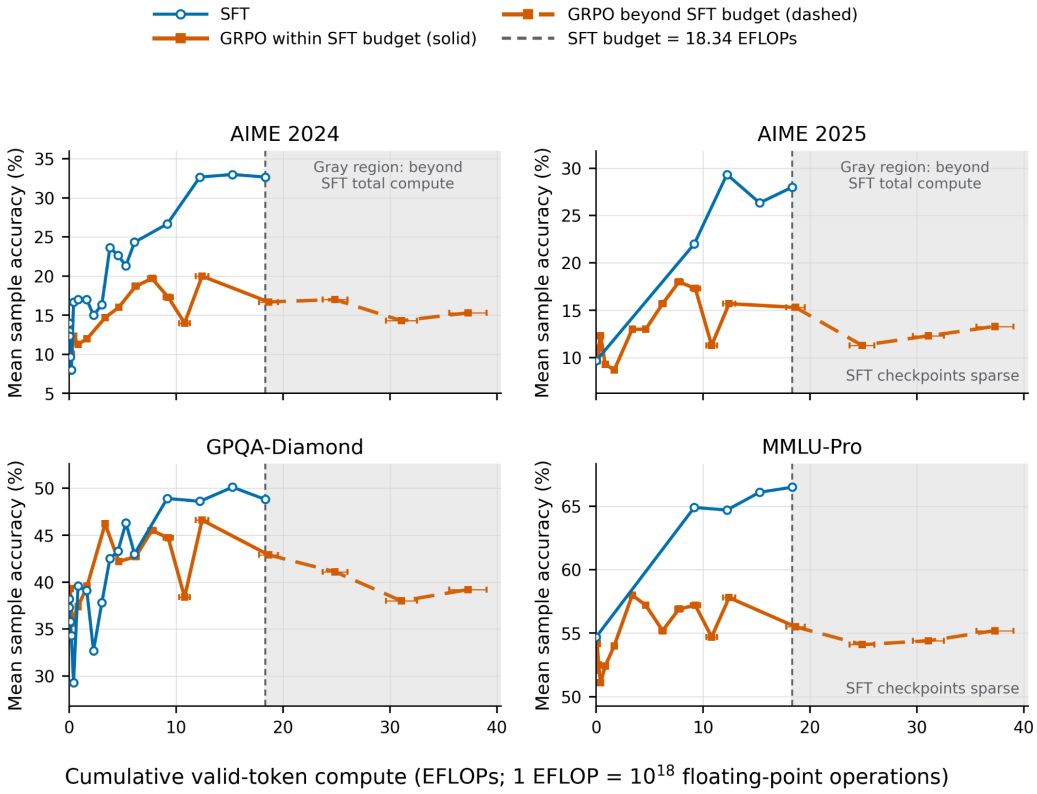


Figure E.1: Mean sample accuracy as a function of cumulative valid-token training compute, where 1 EFLOP $= 10^{18}$ floating-point operations. The vertical gray line marks the complete SFT budget of 18.34 EFLOPs. Orange solid GRPO segments fall within this budget, whereas orange dashed segments denote the continuation of the same GRPO run beyond it and therefore have no observed equal-compute SFT checkpoint. The gray region marks the same beyond-budget range. GRPO horizontal intervals represent sensitivity to the observed 520K–570K trajectory-token range and are not confidence intervals. Because historical SFT evaluations used different sampling contracts, all panels report mean per-sample accuracy rather than relabeling legacy results as pass@8.