ON-POLICY OR OFF-POLICY LEARNING? A SYSTEMATIC STUDY OF DISTILLATION DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy learning has been argued to reduce catastrophic forgetting, produce sparser parameter updates, and improve generalisation. However, existing comparisons between supervised fine-tuning and reinforcement learning vary many factors simultaneously, making the contribution of rollout policy difficult to isolate. We study the effect of rollout policy in a controlled strong-to-weak distillation setting, by independently varying rollout policy, token-level KL direction, and learning rate across the Llama 3 and Qwen2.5 model families and reasoning tasks spanning scientific, medical, and arithmetic domains. Our analysis reveals a nuanced picture of distillation dynamics in which rollout policy does not necessarily play a central role. Instead, KL direction more clearly shapes task performance and output coverage, while learning rate governs forgetting and sparsity. Token-level gradient analysis and experiments along a continuous student–teacher rollout-policy spectrum explain this pattern: forward KL is remarkably robust to rollout policy, whereas reverse KL is substantially more sensitive and favours student-generated rollouts. On-policy data nevertheless improves generalisation to harder Countdown variants under both KL directions, although this advantage does not reliably persist after subsequent RLVR. Our broader conclusions remain robust to removing gradient clipping, using sampled KL estimators, and training on tasks requiring longer reasoning chains. Overall, our results challenge the view that on-policy rollouts are inherently preferable and show that their value depends critically on the objective, evaluation setting, and optimisation hyperparameters.

1 INTRODUCTION

Post-training plays a central role in developing the reasoning capabilities of large language models across a wide range of domains. Prominent approaches include supervised fine-tuning (SFT) on high-quality reasoning traces (Guha et al., 2025; Bai et al., 2025), reinforcement learning with verifiable rewards (RLVR) (DeepSeek-AI, 2025; Shao et al., 2024; Yu et al., 2025), and knowledge distillation (KD) (Hinton et al., 2015; Agarwal et al., 2023; Xiao et al., 2026). Strong-to-weak distillation has become especially prominent in modern post-training pipelines, sing stronger teacher models to transfer reasoning behaviour to weaker students (Yang et al., 2025a; DeepSeek-AI et al., 2026). As the number of available methods grows, classifying their relative strengths and limitations, becomes essential, guiding the development of more effective and efficient training recipes.

One increasingly influential hypothesis singles out the policy used to generate training data as a key factor differentiating post-training methods. Off-policy methods, such as SFT, learn from a fixed dataset or data-generating policy, whereas on-policy methods, such as RLVR, repeatedly train on outputs sampled from the model being optimised. Although SFT and RLVR differ in several other respects, this difference in rollout policy has been proposed as a central factor underlying their differing behaviour. In particular, relative to off-policy learning, on-policy learning has been argued to reduce or prevent catastrophic forgetting (Shenfeld et al., 2025; Chen et al., 2025; Shenfeld et al., 2026), produce substantially sparser parameter updates (Mukherjee et al., 2025), and improve generalisation (Chu et al., 2025; Zhang et al., 2026b; Yuan et al., 2026; Ming et al., 2025). These findings have contributed to broader interest in on-policy post-training, including on-policy distillation (Song and Zheng, 2026; Agarwal et al., 2023; Lu and Lab, 2025; Xiao et al., 2026).

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.

Yet the causal role of rollout policy remains unclear. Comparisons between SFT and RLVR also change the objective, source and density of supervision, and optimisation procedure, so effects attributed to on-policy data may arise from these accompanying differences. Resolving this ambiguity is practically important: on-policy methods require continual generation throughout training, whereas off-policy methods can reuse existing data. If their proposed benefits arise from other design choices, this additional cost may often be unnecessary.

In this work, we use strong-to-weak distillation as a controlled testbed for isolating the effect of rollout policy. Unlike SFT vs RLVR comparisons, this setting allows us to change the data-generating policy while holding the remaining training pipeline fixed. Across the Llama 3 and Qwen2.5 model families and reasoning tasks spanning scientific, medical, and arithmetic domains, we independently vary the rollout policy and KL direction, breaking the conventional pairing of forward KL with off-policy learning and reverse KL with on-policy learning, while controlling for the learning rate. We find *no consistent advantage from on-policy distillation* in final in-distribution accuracy, catastrophic forgetting, or parameter-update sparsity. Instead, KL direction more clearly determines task accuracy, while learning rate governs forgetting and sparsity. In particular, **we find no evidence that on-policy rollouts intrinsically preserve prior capabilities or induce sparser updates.**

To understand when rollout policy does matter, we examine its interaction with the token-level KL objective through a theoretical and empirical analysis. We vary the rollout policy along a student–teacher spectrum, interpolating between the two policies and extrapolating beyond them to favour tokens preferred by one model over the other. Our analysis reveals a clear objective-dependent asymmetry: forward KL is remarkably robust to changes in the rollout policy, whereas reverse KL is substantially more sensitive and favours on-policy rollouts. Nevertheless, *forgetting and update sparsity remain governed by learning rate*, with little to no variation caused by the rollout policy.

The benefits of different configurations become more nuanced when considering evaluations beyond final accuracy. Forward KL produces larger pass@k gains on the in-distribution task compared to reverse KL, while **generalisation to harder task variants on Countdown consistently benefits from on-policy data**, under both KL directions. However, this initial generalisation advantage of on-policy learning does not reliably persist after subsequent RLVR. When combined with reverse KL, on-policy rollouts also reduce incidental teacher-style transfer. Our principal findings remain robust when removing gradient clipping, replacing full-vocabulary KL with commonly used sampled estimators, and training on tasks requiring longer reasoning chains. **Our main contributions are:**

- **A controlled study of distillation.** We disentangle rollout policy, KL direction, and learning rate in a controlled study of strong-to-weak distillation, finding no consistent advantage from on-policy rollouts in final accuracy, catastrophic forgetting, or parameter-update sparsity.
- **An objective-dependent account of rollout-policy sensitivity.** Through gradient analysis and experiments along a continuous student–teacher spectrum, we show that forward KL is robust to rollout policy, whereas reverse KL is substantially more sensitive.
- **A characterisation of when on-policy data helps.** While training stability and output coverage are largely driven by KL direction, and learning rate determines the degree of catastrophic forgetting and update sparsity, we find that on-policy rollouts consistently help to improve model's generalisation to harder task variants and could reduce incidental teacher-style transfer.

Together, these findings challenge the view that on-policy rollouts are universally preferable for strong-to-weak distillation. Instead, the choice of token-level KL divergence strongly affects task performance, training stability and output coverage, while catastrophic forgetting and update sparsity are driven by the learning rate. We hope that these findings encourage practitioners to weight the additional cost of on-policy distillation over cheaper off-policy alternatives, and help clarify which training outcomes can be attributed to rollout policy rather than KL direction or learning rate.

2 Related Works

Strong-to-Weak and On-Policy Distillation. Knowledge distillation transfers capabilities from a teacher to a student by matching their output distributions (Hinton et al., 2015; Kim and Rush, 2016). In language-model post-training, strong-to-weak distillation has become a common approach for transferring reasoning capabilities from larger models to smaller ones (Abdin et al., 2024; Xiao et al.,

2026; Yang et al., 2025a; DeepSeek-AI et al., 2026; Bai et al., 2026). The rollout source defines an important distinction within these methods: in OffPD, the student trains on fixed teacher-generated responses (Sanh et al., 2020), while in OnPD, the student generates trajectories and queries the teacher for token-level supervision along them (Agarwal et al., 2023). Hybrid variants interpolate between the two, for example by constructing trajectories from mixtures of student- and teacher-generated tokens (Xu et al., 2024). The growing literature on OnPD has investigated its algorithmic variants and failure modes (Song and Zheng, 2026; Li et al., 2026b; Armandpour et al., 2026; Jia et al., 2026; Li et al., 2026a; Zhu et al., 2026), but few works directly contrast OnPD with the cheaper OffPD alternative while holding the remaining training configuration fixed.

Claims About On-Policy Post-Training. In the broader post-training literature, recent studies have attributed several advantages to on-policy training, such as better preservation of previously acquired capabilities, thereby reducing catastrophic forgetting (Shenfeld et al., 2025; Chen et al., 2025; Shenfeld et al., 2026; Lu and Lab, 2025), sparser parameter updates (Mukherjee et al., 2025; Yu et al., 2026) and improved generalisation beyond the training distribution (Chu et al., 2025; Yuan et al., 2026; Zhang et al., 2026b; Ming et al., 2025). However, much of this evidence compares SFT with RLVR, analyses public checkpoints, or otherwise varies multiple aspects of the training recipe simultaneously. Such comparisons can entangle rollout source with the learning objective, reward signal, supervision density, optimisation scale, use of gradient clipping, and learning rate, which has itself been shown to strongly affect learning–forgetting trade-offs (Catalan-Tatjer and Geiping; Rofin et al., 2026). We therefore treat these reported results as hypotheses about the advantages of on-policy learning and test them in a strong-to-weak distillation setting where the rollout policy, KL direction, and learning rate are varied explicitly.

3 Background

Notation. For the language-modelling tasks considered in this work, let x and y denote the input and output sequences, respectively, each consisting of tokens from the vocabulary $\mathcal{V}$ with $|\mathcal{V}| = M$. Let $y_{<n} = (y_1, \ldots, y_{n-1})$ denote the output prefix preceding the n^{th} token, and let L_y denote the length of y. We assume that the input sequences (prompts) are sampled from a fixed data distribution p_{data}, while the output sequences are generated autoregressively from a policy π, which for a given input x and a partially generated output $y_{<n}$ outputs a discrete distribution over the entire token vocabulary $\mathcal{V}$, $\pi(\cdot|x, y_{<n}) \in \Delta(\mathcal{V})$.

Strong-to-Weak Distillation. In strong-to-weak knowledge distillation we assume access to two autoregressive language models: a student π_S^θ (parametrised by θ) and a teacher π_T providing the supervision signal. Let ρ denote the rollout policy used to generate the output sequences on which distillation is performed. Then, the distillation objective can be defined as:

$$\mathcal{L}(\theta) = \mathbb{E}_{x\sim p_{\text{data}}} \mathbb{E}_{y\sim\rho(\cdot|x)} \left[\frac{1}{L_y} \sum_{n=1}^{L_y} \mathcal{D}\left(\pi_S^\theta(\cdot|x, y_{<n}) \,||\, \pi_T(\cdot|x, y_{<n})\right) \right], \tag{1}$$

where $\mathcal{D}$ is a distance function quantifying the discrepancy between the next-token student and teacher distributions. The objective therefore averages the token-level discrepancy over prefixes visited under the rollout policy ρ.

Under this formulation, we obtain **off-policy distillation** (OffPD) by setting $\rho = \pi_T$ and **on-policy distillation** (OnPD) by setting $\rho = \pi_S^\theta$. Consequently, in OnPD the loss function depends on θ also through the trajectory sampling procedure $y \sim \pi_S^\theta(\cdot|x)$. However, for computational tractability, we stop gradients through the sampling process and treat sampled trajectories as fixed when computing each update (Agarwal et al., 2023; Tang and Munos, 2025).

KL Divergence as the Distance Function. We use the Kullback–Leibler divergence to measure the distance between the student and teacher next-token distributions. Because we have access to their full output distributions, we compute the divergence over the entire vocabulary. This avoids the additional variance introduced by estimating the token-level KL from sampled tokens and uses the complete supervision signal provided by the teacher (DeepSeek-AI et al., 2026). Given that the KL divergence is not symmetric, we can distinguish between the forward and the reverse KL:

$$\text{Forward-KL:} \quad D_{\text{F-KL}}(\pi_S, \pi_T) = \mathbb{E}_{y_n\sim\pi_T} \left[\log \frac{\pi_T(y_n)}{\pi_S(y_n)} \right] = \sum_{v\in\mathcal{V}} \pi_T(v) \log \frac{\pi_T(v)}{\pi_S(v)}, \tag{2}$$

$$\text{Reverse-KL:} \quad D_{\text{R-KL}}(\pi_S, \pi_T) = \mathbb{E}_{y_n \sim \pi_S}\left[\log \frac{\pi_S(y_n)}{\pi_T(y_n)}\right] = \sum_{v \in \mathcal{V}} \pi_S(v) \log \frac{\pi_S(v)}{\pi_T(v)}. \tag{3}$$

For conciseness, we write $\pi(y_n) = \pi(\cdot|x, y_{<n})$ for the distribution at the n^{th} token. Forward KL places greater weight on tokens to which the teacher assigns substantial probability and strongly penalises the student for failing to cover them. It is therefore commonly described as *mode-covering*. Reverse KL in turn places greater weight on tokens favoured by the student, and penalises assigning probability to tokens which are unlikely under the teacher, giving rise to its *mode-seeking* behaviour.

Rollout Policy and KL Direction Coupling. Although OnPD is often associated with reverse KL and OffPD with forward KL, the rollout policy and token-level KL direction are conceptually distinct. The rollout policy ρ determines which prefixes are visited, while the KL direction determines how the student and teacher distributions are compared at those prefixes. The conventional pairings have two motivations. Firstly, when considering the sequence-level KL divergence between the student and the teacher, teacher rollouts paired with forward KL and student rollouts paired with reverse KL are exactly the two chain-rule decompositions of sequence-level KL (see AppendixC.5). However, because we stop gradients through student-sampled prefixes, this pairing holds only at the objective-value level: the OnPD reverse-KL update is generally *not* the total gradient of sequence-level reverse KL. Second, at a fixed prefix, $\log[\pi_T(v)/\pi_S^\theta(v)]$ with $v \sim \pi_T$ is an unbiased one-sample estimator of forward KL, while $\log[\pi_S^\theta(v)/\pi_T(v)]$ with $v \sim \pi_S^\theta$ is an unbiased estimator of reverse KL. When token-level KLs are estimated from single samples, the rollout tokens therefore naturally pair OffPD with forward KL and OnPD with reverse KL. In our setting, however, we compute the token-level KL *over the full vocabulary* $\mathcal{V}$, so the KL direction does not constrain the rollout policy. Moreover, neither the chain-rule identities nor the sampling interpretation implies that the conventional pairings are easier to optimise or outperform the crossed pairings. We therefore treat rollout policy and KL direction as independent design choices and study their interaction experimentally.

4 A Controlled Comparison of On- and Off-Policy Distillation

4.1 Experimental Setup

Experimental design. We compare OnPD and OffPD while independently varying token-level KL direction and learning rate (1×10^{-5} or 5×10^{-5}). We compute both KL objectives over the full vocabulary and keep the remaining optimisation settings fixed. Each student is trained with full-parameter fine-tuning for 150 optimisation steps, with three random seeds per configuration. Complete training details are provided in Appendix D.3.

Models and tasks. Our main experiments distil Llama-3.1-8B teachers into Llama-3.2-1B students (Grattafiori et al., 2024), while Appendix A.5 reports corresponding experiments with Qwen2.5-7B teachers and Qwen2.5-1.5B students (Yang et al., 2025b). We consider three reasoning tasks spanning different domains: *MedReason* (Wu et al., 2025) for medical reasoning, *Science* (Feng et al., 2024) for scientific reasoning, and *Countdown* (Pan et al., 2025) for arithmetic reasoning. Teachers and students are taken from the same model family to ensure a shared tokenizer and vocabulary, allowing to directly compare their token-level distributions. Since overlapping capabilities within a model family can limit knowledge transfer (Li et al., 2026b), we train a separate teacher on each task to establish a sufficient capability gap, then freeze it throughout distillation. Dataset preparation and teacher training are detailed in Appendix D.1 and D.2.2.

Evaluation. For each trained checkpoint, we assess held-out accuracy on the target task, catastrophic forgetting, and parameter-update sparsity. We measure catastrophic forgetting by the decrease in the mean score across seven out-of-distribution benchmarks from before to after training (Appendix D.1.2). Following Mukherjee et al. (2025), we measure update sparsity as the fraction of parameters whose absolute change from the initial checkpoint is below 10^{-6} (Appendix D.4).

4.2 Results

Final task performance. On-policy rollouts offer no consistent advantage in target-task accuracy: the best mean accuracies across the three datasets are nearly identical, reaching 72% for OnPD and 73% for OffPD (Figure 1, left). KL direction produces a clearer pattern: forward KL achieves

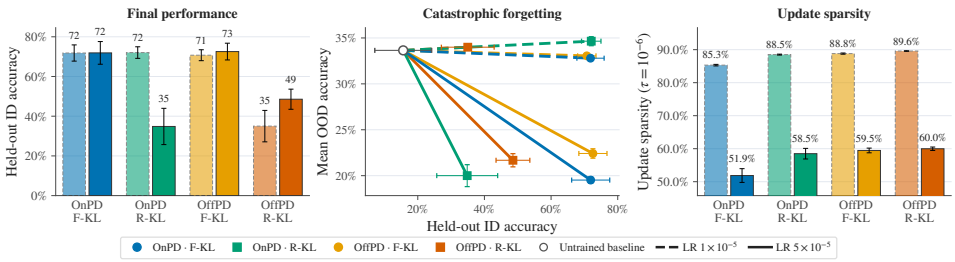


Figure 1: **Comparison of OnPD and OffPD across three reasoning tasks.** Held-out ID accuracy in the left and middle panels is averaged across MedReason, Science, and Countdown-3. Per-dataset results are reported in Figure 20. Error bars show SEM computed over $N = 3$ seeds.

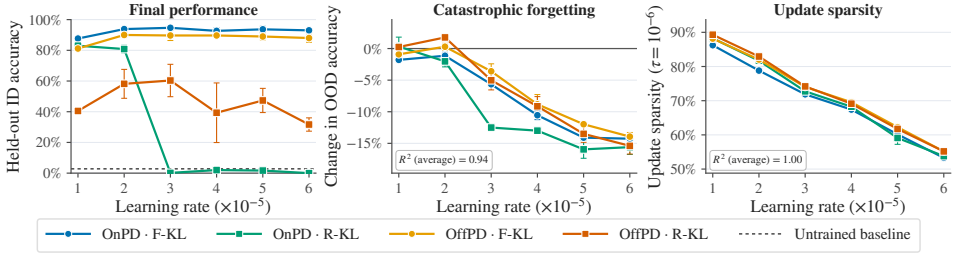


Figure 2: **The effect of learning rate on Countdown-3.** Forgetting increases and update sparsity decreases with the learning rate under both rollout policies. Points report means over $N = 3$ seeds; error bars show SEM.

71–73% across all rollout policies and learning rates, whereas reverse KL ranges from 35% to 72% and is substantially more sensitive to the learning rate.

Catastrophic forgetting. Similar task performance might conceal different amounts of forgetting on OOD tasks. If on-policy rollouts mitigate forgetting, as suggested in prior work (Shenfeld et al., 2025; Chen et al., 2025), we would expect OnPD to preserve OOD performance better than OffPD with the same KL objective and learning rate. Instead, mean OOD performance changes by at most 1.3 percentage points at the lower learning rate but drops by 11.2–14.0 points at the higher rate (Figure 1, middle). Rollout-policy differences are modest compared to that. Remarkably, with forward KL, lowering the learning rate mitigates most of this forgetting while maintaining comparable ID accuracy. In this setting, learning rate, rather than on-policy rollouts, is the dominant factor.

Parameter-update sparsity. The same separation by learning rate appears in the parameter updates (Figure 1, right). Sparsity ranges from 85.3–89.6% at the lower learning rate, compared with 51.9–60.0% at the higher learning rate. Again, differences between rollout policies are much smaller: OffPD produces at least as sparse updates as OnPD in every matched comparison. KL direction has a secondary effect, with reverse KL generally producing greater sparsity than forward KL on on-policy rollouts. Thus, sparser updates do not emerge as a benefit of on-policy rollouts in these experiments.

Learning-rate sweep. The comparison above uses two learning rates. To examine these effects more systematically, we sweep the learning rate from 1×10^{-5} to 6×10^{-5} on Countdown-3, keeping the remaining training configuration fixed. Figure 2 shows that update sparsity decreases approximately linearly as learning rate increases, while OOD performance also deteriorates consistently, with similar trends under both rollout policies. At matched learning rates, OffPD exhibits less forgetting in nearly every comparison, although this difference remains smaller than the effect of learning rate. Crucially, **greater forgetting does not appear to be a necessary cost of achieving strong target-task performance**: under forward-KL, using small learning rate allows to achieve top performance, without leading to a decrease in the prior capabilities. Models with similar final task performance can therefore exhibit markedly different degrees of forgetting, resembling the "cliff" phenomenon previously identified in the SFT setting by Catalan-Tatjer and Geiping.

Takeaways. Across our controlled comparisons, on-policy rollouts offer no consistent advantage in in-distribution performance, forgetting, or update sparsity. KL direction more clearly distinguishes task performance, while learning rate largely governs forgetting and sparsity.

5 When does rollout policy actually matter?

Our results in Section 4 indicate that when controlling for the direction of the KL divergence and the learning rate, the differences between OnPD and OffPD seem negligible, challenging the view that rollout policy strongly affects performance. This inspires a broader question: *when, if at all, does the rollout policy actually matter*? A closer look at Figure 1 reveals that the answer might depend on the direction of the token-level KL divergence. Indeed, in this section we carry out a controlled analysis which reveals that forward KL is robust to the changes in the rollout policy, while for reverse KL rollout policy matters a lot, with student-favoured trajectories leading to better performance.

5.1 Logit Gradients Reveal Differing Sensitivity to the Rollout Policy

To understand how the rollout policy affects optimisation, we analyse the token-level gradients of forward and reverse KL. Let $(z_S^\theta)_v$, $v \in \mathcal{V}$ denote the logit values produced by student model at a fixed prefix, with $\pi_S^\theta(v) = \mathrm{softmax}(z_S^\theta)_v$. Then, we can describe the parameter gradients as follows:

$$\nabla_\theta D_{\text{F-KL}} = \sum_{v \in \mathcal{V}} (\pi_S^\theta(v) - \pi_T(v)) \nabla_\theta (z_S^\theta)_v, \tag{4}$$

$$\nabla_\theta D_{\text{R-KL}} = \sum_{v \in \mathcal{V}} \pi_S^\theta(v) \left[\log \frac{\pi_S^\theta(v)}{\pi_T(v)} - D_{\text{R-KL}} \right] \nabla_\theta (z_S^\theta)_v. \tag{5}$$

We provide the derivation in Appendix B. These expressions reveal an important asymmetry. The forward-KL derivative with respect to the student logits is $\pi_S^\theta(v) - \pi_T(v)$; it is therefore nonzero whenever the teacher and student next-token distributions differ, and each of its coordinates lies in $[-1, 1]$. Consequently, under bounded student-logit Jacobians $\nabla_\theta(z_S^\theta)$, the difference between the forward-KL updates induced by two rollout policies is bounded linearly by the total-variation distance between the prefix distributions they induce (see Appendix C for the precise statement and proof). Hence, small changes in the trajectories generated by the rollout policy produce proportionally small changes in the forward-KL gradient.

For reverse KL, however, even a small rollout change can produce an arbitrarily large gradient change when the teacher and student assign very different probabilities to some tokens. Its derivative with respect to student logits is weighted by $\pi_S^\theta(v)$, so it vanishes as the student probability approaches zero, even when the teacher assigns the token substantial probability. Reverse KL may therefore struggle to recover teacher modes omitted by the student, reflecting its mode-seeking behaviour. Conversely, when the student assigns appreciable probability to a token that the teacher considers extremely unlikely, the log-ratio $\log \pi_S^\theta(v)/\pi_T(v)$ an become arbitrarily large, potentially producing sharp, high-variance updates, which can potentially destabilise training. Accordingly, unlike forward KL, reverse KL admits no bound dependent solely on the rollout distance and the student-logit Jacobian. We formalise this result in Appendix C.

These properties suggest that reverse KL is more sensitive to the rollout policy. While forward KL provides signal at any visited prefix where the policies disagree, reverse KL emphasises student-supported, teacher-disfavoured tokens and therefore depends more strongly on visiting the student's own prefix distribution. We consequently expect reverse KL to benefit more from on-policy rollouts.

5.2 Empirical Validation Using a Rollout-Policy Spectrum

Rollout-policy spectrum. To empirically validate the differing sensitivity of forward and reverse KL to the rollout policy, we systematically vary the rollout policy along a student–teacher spectrum parametrised by λ. Specifically, we define a likelihood-ratio-controlled policy π_λ, where, for each $v \in \mathcal{V}$, $\pi_\lambda(v) = \mathrm{softmax}(z_\lambda)_v$, with

$$(z_\lambda)_v = \frac{1}{2} \left(\log \pi_S(v) + \log \pi_T(v) \right) + \frac{\lambda}{2} \left(\log \pi_S(v) - \log \pi_T(v) \right).$$

Negative values of λ produce teacher-favoured rollouts, positive values produce student-favoured rollouts, and $\lambda = 0$ is the symmetric midpoint. In particular, $\lambda = -1$ recovers standard OffPD, while $\lambda = 1$ recovers OnPD. Choosing $\lambda < -1$ biases sampling towards tokens which the teacher finds plausible but the student does not, and vice versa for $\lambda > 1$. To avoid aggressively amplifying

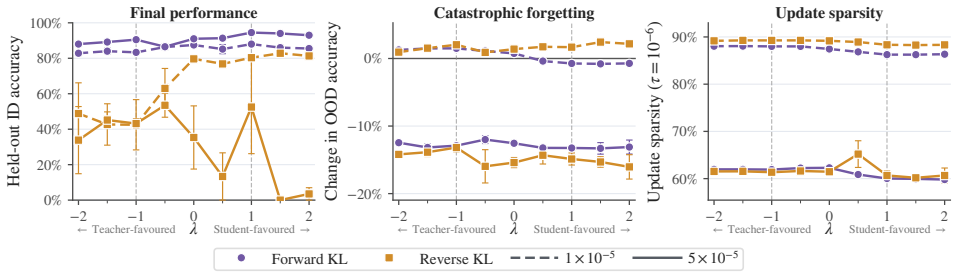


Figure 3: **Performance of forward and reverse KL across the rollout-policy spectrum.** Forward KL maintains strong final performance across the spectrum, whereas reverse KL varies substantially. Rollout policy has comparatively little effect on catastrophic forgetting or update sparsity. Points show means over $N = 3$ seeds; error bars denote SEM.

tokens that both models consider unlikely, we additionally employ log-ratio clipping and plausibility masking, similar to Li et al. (2023), as detailed in Appendix D.5. Figure 4 validates that this sampling scheme allows us to vary the student and teacher likelihoods on the generated trajectories.

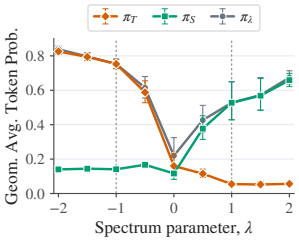


Figure 4: **Likelihood in the rollout-policy spectrum**.

Empirical results. Following the setup of the previous section, we train the student on Countdown-3 for 150 optimisation steps using either forward or reverse KL objective, with trajectories sampled from π_λ. The results in Figure 3 support the conclusions from our gradient analysis. Forward KL is remarkably robust to the rollout policy: held-out accuracy remains above 80% across the entire spectrum, varying by only 5.2 percentage points at a learning rate of 1×10^{-5}. Remarkably, OnPD with forward KL can learn effectively despite passing through a regime of largely incoherent, high-entropy rollouts (Appendix A.10). Similar final performance under OnPD and OffPD may therefore conceal markedly different optimisation paths. Reverse KL is substantially more sensitive, exhibiting large changes in mean performance across λ and considerable variance across seeds for several settings. At the lower learning rate, reverse KL benefits strongly from student-favoured rollouts ($\lambda > 0$), while performance deteriorates for teacher-favoured rollouts. At the higher learning rate, however, reverse KL remains unstable and occasionally collapses.

We further examine catastrophic forgetting and update sparsity over the same rollout policy spectrum. Middle and right panels of Figure 3 show that varying the rollout policy has a comparatively modest effect on mean OOD performance and sparsity. Both are influenced much more strongly by the learning rate, consistent with our earlier findings.

> **Takeaways.** Gradient analysis and empirical experiments reveal that forward KL is remarkably robust to the rollout policy, while reverse KL favours student-generated rollouts. Catastrophic forgetting and update sparsity are only marginally affected by rollout policy.

5.3 Are there any other differences induced by the direction of KL divergence?

So far, we have seen that the effect of rollout policy on task performance depends strongly on KL direction. However, pass@1 accuracy alone does not fully capture what a model has learned. We therefore examine output coverage under repeated sampling, generalisation to harder Countdown variants, and performance under subsequent RLVR.

Forward KL yields larger gains from repeated sampling. Improvements in pass@1 do not necessarily translate into improvements at higher pass@k (Yue et al., 2025). Figure 5 (left pair) compares pass@1 and pass@10 on Countdown-3 along the rollout-policy spectrum with learning 1×10^{-5} (LR 5×10^{-5} shown Figure 12). Reverse KL yields smaller gains at comparable pass@1: for $\lambda > 0$ (OnPD), both KL directions have similar pass@1, but forward KL leads to significantly higher pass@10. Results on Qwen2.5 show the same effect across the rollout-policy spectrum (Figure 16). This suggests that KL direction influences output coverage more than rollout policy.

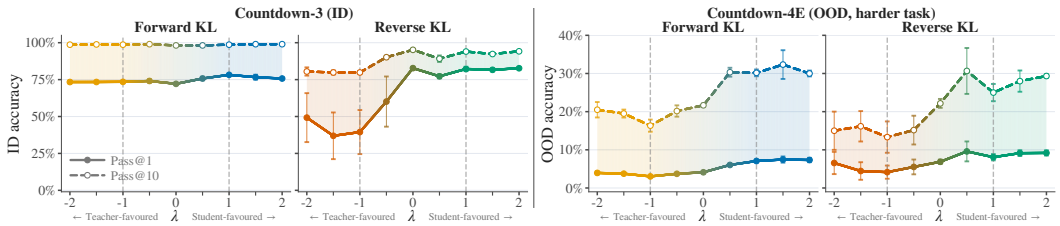


Figure 5: **Output coverage and generalisation along the rollout-policy spectrum.** While in-distribution forward KL leads to better output coverage, when generalising to harder tasks we can see a clear benefit of using on-policy data. Error bars show SEM across three seeds. The easier Countdown-3 uses $T = 1$, while the more difficult Countdown-4E uses $T = 0.5$.

More on-policy rollouts generalise better to harder tasks. We evaluate the same Countdown-3 checkpoints without further training on the harder task Countdown-4E, which uses four operands drawn from 1–10. Figure 5 (right pair) shows pass@1 and pass@10 along the rollout-policy spectrum with learning 1×10^{-5} (LR 5×10^{-5} shown Figure 12). For both KL directions, performance increases gradually across the spectrum, with more on-policy rollouts ($\lambda > 0$) consistently yielding $10 - 15\%$ higher pass@k accuracy than off-policy rollouts. Results on Qwen2.5 with the same LR show consistent trends. Hence, by investigating generalisation to harder tasks, we identify the first setting where using on-policy rollouts leads to a significant performance improvement compared to off-policy rollouts. These findings are consistent with prior works studying generalisation in the context of SFT and RL (Yuan et al., 2026; Zhang et al., 2026b; Ming et al., 2025).

Consequences for subsequent RLVR. Which configurations, then, provide better starting points for further RLVR? We train the Countdown-3 distillation checkpoints on Countdown-4 for 300 RLVR steps, using three seeds per distillation configuration and an otherwise fixed RL setup. Figure 6 shows that reverse-KL checkpoints distilled at a learning rate of 1×10^{-5} improve quickly—including OffPD checkpoints with initially low accuracy—but later undergo reward collapse. In contrast, OffPD checkpoints distilled with forward KL at 1×10^{-5} or reverse KL at 5×10^{-5} (Appendix Figure 26) improve steadily and reach the highest performance at the end of training, despite starting near 0% accuracy. The strongest sustained RLVR performance therefore comes from off-policy checkpoints, despite their lower initial Countdown-4 accuracy. The initial generalisation advantage of on-policy checkpoints does not translate into a reliable advantage after RLVR. These findings are consistent with Zhang et al. (2026a), who show that the SFT checkpoints with the strongest initial performance are not necessarily the best starting points for a subsequent RLVR stage, suggesting that additional research is required to determine how to best combine post-training methods into multi-step pipelines.

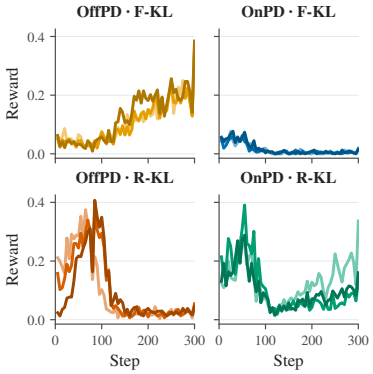


Figure 6: **RLVR rewards on Countdown-4** when starting from distillation checkpoints trained with LR 1×10^{-5}; each curve corresponds to a distillation seed.

Teacher-style transfer. Beyond output coverage and generalisation, we proceed to study one more aspect of training: the transfer of incidental teacher behaviours. We instruct the teacher to reason in Spanish and measure how often the student subsequently produces Spanish responses (Section A.11). Surprisingly, OnPD with reverse KL largely preserves the student's English response style, whereas the other configurations exhibit near-complete transfer to Spanish. We hypothesise that, on student-generated English prefixes, the teacher still supports English continuations despite its Spanish instruction. The mode-seeking reverse-KL objective can therefore match this English mode without requiring the student to cover Spanish alternatives. These results suggest that combining OnPD with reverse KL may help suppress incidental style transfer.

Takeaways. While forward KL improves in-distribution output coverage, when generalising to harder tasks we observe consistent benefits from using on-policy data, across both forward and reverse KL. However, this advantage of OnPD does not reliably persist after RLVR.

6 Do these findings generalise beyond our experimental design?

We test whether the findings in the preceding sections depend on three choices in our experimental design: computing KL over the full vocabulary rather than sampled tokens, applying gradient clipping, and studying tasks with relatively short rollouts.

Sampled rather than full-vocabulary KL. Our main experiments compute the KL divergence over the full vocabulary, since this has been shown to improve training stability (DeepSeek-AI et al., 2026). Several recent works instead use a more memory-efficient sampled-KL estimator (Lu and Lab, 2025; Li et al., 2026b). We therefore repeat our experiments using sampled KL. As in our main analysis, we disentangle the policy used to generate the rollout from the distribution used to sample the token-level KL gradient, evaluating all four combinations of rollout policy and KL direction. In the results, forward KL remains robust to rollout policy, reverse KL remains substantially more sensitive and brittle, and learning rate continues to govern forgetting and update sparsity (Appendix A.2).

Removing gradient clipping. Our main experiments clip the global gradient norm at 1.0, and the unclipped norm exceeds this threshold throughout training (Figure 7). Clipping could therefore suppress meaningful differences in gradient magnitudes between rollout policies. Disabling clipping exposes a stronger difference between KL directions but no consistent OnPD–OffPD gap. Reverse-KL performance deteriorates substantially, whereas forward KL remains effective at larger learning rates. Forgetting and sparsity remain primarily determined by learning rate (Appendix A.1). Thus, clipping stabilises reverse KL but does not explain the limited effect of rollout policy.

Training on datasets requiring longer rollouts. Our main experiments cover medical, scientific, and arithmetic reasoning, but all three have average teacher-response lengths of 94–135 tokens (Table 2). The benefits of on-policy distillation may become more apparent over longer trajectories, where small differences between the student and teacher policies can compound across many generated tokens. We therefore repeat the comparison using Qwen2.5 models on Numina–MATH, where teacher responses average 622 tokens. At the lower learning rate, OnPD with reverse KL achieves the highest MATH-500 accuracy among the evaluated configurations, suggesting that on-policy rollouts may help in harder tasks requiring longer reasoning. Nevertheless, forward KL remains more robust than reverse KL, while forgetting and sparsity remain governed by learning rate (Appendix A.3). Because this experiment contains one run per condition, we treat the OnPD advantage as suggestive.

> **Takeaways.** Our main conclusions are not explained by full-vocabulary KL, gradient clipping, or short rollouts: forward KL remains more robust to rollout policy, while learning rate remains the strongest predictor of forgetting and parameter-update sparsity.

7 Discussion, Limitations and Future Work

Discussion. In conclusion, our controlled comparison offers a nuanced view of the differences between on-policy and off-policy learning in the context of distillation. While on-policy learning can improve generalisation to harder tasks, and potentially reduce incidental teacher-style transfer, learning rate and KL direction explain substantially more of the observed variation. Given its lower computational cost, we encourage future work on on-policy distillation to include OffPD as a standard baseline. More broadly, our results suggest that it is difficult to attribute most of the observed differences between SFT and RL to rollout policy alone, motivating closer study of other factors such as the learning objective, reward signal, supervision density, and optimisation procedure.

Limitations and Future Work. Our controlled experimental setting enables systematic comparisons across rollout policies, KL directions, and learning rates, but necessarily focuses on a bounded regime: student models of at most 1.5B parameters, and (consequently) tasks requiring reasoning traces of at most 2,000 tokens. Extending this analysis to larger models, and tasks requiring longer rollouts represents a promising direction for future work. Further, while in our study we keep the teacher model fixed, in future work we would like to investigate whether differences between OnPD and OffPD emerge under specific student–teacher combinations.

AI USE STATEMENT

In this work, we used generative AI tools to obtain demonstrations for solving reasoning tasks, to help formulate mathematical claims, to provide guidelines for proving those claims and assist in the writing of proofs, to implement and maintain the code for running experiments, and to prepare plots for the figures. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, to propose or refine hypotheses, to design our research methodology, to support qualitative and thematic data analysis, to suggest a structure for the paper, or to interpret results, and the generation of synthetic datasets and assistance with translation are not applicable to this work. Additionally, we have used generative AI tools to edit individual paragraphs in the paper to improve readability. We have reviewed all AI-assisted work: all mathematical claims and their proofs were checked in full by the authors, AI-assisted code was verified and tested for correctness by two authors, and all AI-edited text was reviewed by all the authors to confirm that it accurately reflects our intended meaning and claims. The plotted values were verified with the matching plots from wandb. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide complete definitions of OnPD, OffPD, forward and reverse KL, and the rollout-policy spectrum, including derivations and implementation details for sampled KL estimation, plausibility masking, and log-ratio clipping. The appendix documents the model checkpoints, datasets, prompts, data splits, generation settings, optimisation hyperparameters, learning rates, gradient-clipping choices, training durations, and evaluation procedures used in each experiment. Unless otherwise stated, results are averaged over three random seeds and reported with the standard error of the mean; experiments containing fewer runs, including the long-rollout ablation, are explicitly identified. All matched comparisons use the same training and evaluation pipeline, differing only in the factors under investigation.

REFERENCES

M. Abdin, S. A. Jacobs, A. Awan, J. Aneja, A. Awadallah, H. Awadalla, N. Bach, A. Bahree, A. Bakhtiari, H. S. Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

R. Agarwal, N. Vieillard, Y. Zhou, P. Stanczyk, S. Ramos, M. Geist, and O. Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes, 2023. URL http://arxiv.org/abs/2306.13649.

M. Armandpour, F. Ilhan, D. Harrison, A. Jaiswal, D. N. M. Hoang, F. Faghri, Y. Zhang, M. Cho, and M. Farajtabar. Unmasking On-Policy Distillation: Where It Helps, Where It Hurts, and Why, 2026. URL http://arxiv.org/abs/2605.10889.

K. T. Y. Bai, Y. Bao, Y. Charles, C. Chen, G. Chen, H.-T. Chen, H.-R. Chen, J. Chen, N.-X. Chen, R. Chen, et al. Kimi K2: Open Agentic Intelligence, 2025. URL http://arxiv.org/abs/2507.20534.

K. T. Y. Bai, Y. Bai, Y. Bao, C. M, J. Cai, X.-H. Cai, P. Cao, Y. Cao, Z. Chai, Y. Charles, et al. Kimi k3: Open frontier intelligence. *arXiv*, 2026.

A. Catalan-Tatjer and J. Geiping. Learning-Forgetting Optimality in Supervised Finetuning: A Cliff Perspective. URL https://openreview.net/forum?id=aWl2TIUjxP&referrer=%5Bthe%20profile%20of%20Jonas%20Geiping%5D%28%2Fprofile%3Fid%3D~Jonas_Geiping1%29.

H. Chen, N. Razin, K. Narasimhan, and D. Chen. Retaining by Doing: The Role of On-Policy Data in Mitigating Forgetting, 2025. URL https://arxiv.org/abs/2510.18874v2.

M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. Evaluating large language models trained on code, 2021.

T. Chu, Y. Zhai, J. Yang, S. Tong, S. Xie, D. Schuurmans, Q. V. Le, S. Levine, and Y. Ma. SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training, 2025. URL http://arxiv.org/abs/2501.17161.

G. Cui, L. Yuan, Z. Wang, H. Wang, Y. Zhang, W. Li, B. He, Y. Fan, T. Yu, Q.-X. Xu, et al. Process reinforcement through implicit rewards. *Trans. Mach. Learn. Res.*, 2025. doi: 10.48550/arXiv.2502.01456.

DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv.org*, 645(8081), 2025. ISSN 0028-0836, 1476-4687. doi: 10.48550/arXiv.2501.12948. URL http://arxiv.org/abs/2501.12948.

DeepSeek-AI, A. Xu, B. Lin, B. Xue, B. Wang, B. Xu, B. Wu, B. Zhang, C. Lin, C. Dong, et al. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

K. Feng, K. Ding, W. Wang, X. Zhuang, Z. Wang, M. Qin, Y. Zhao, J. Yao, Q. Zhang, and H. Chen. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv.org*, 2024. doi: 10.48550/arXiv.2406.09098.

L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, A. Le Noac'h, et al. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The Llama 3 Herd of Models, 2024. URL http://arxiv.org/abs/2407.21783.

E. Guha, R. Marten, S. Keh, N. Raoof, G. Smyrnis, H. Bansal, M. Nezhurina, J. Mercat, T. Vu, Z. Sprague, et al. OpenThoughts: Data Recipes for Reasoning Models, 2025. URL http://arxiv.org/abs/2506.04178.

T. Hartvigsen, S. Gabriel, H. Palangi, M. Sap, D. Ray, and E. Kamar. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.234.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network, 2015. URL http://arxiv.org/abs/1503.02531.

J. Hong, N. Lee, and J. Thorne. Orpo: Monolithic preference optimization without reference model, 2024.

N. Jia, H. Yang, X. Ma, J. Lian, S. Zhang, W. Zhang, K. Zeng, X. Cai, and Z. Sun. Asymmetric On-Policy Distillation: Bridging Exploitation and Imitation at the Token Level, 2026. URL http://arxiv.org/abs/2605.06387.

Y. Kim and A. M. Rush. Sequence-level knowledge distillation. In J. Su, K. Duh, and X. Carreras, editors, *Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1139.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014.

W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with pagedattention. In *Symposium on Operating Systems Principles*, pages 611–626. ACM, 2023. doi: 10.1145/3600006.3613165.

N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, S. Lyu, et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv.org*, 2024. doi: 10.48550/arXiv.2411.15124.

J. LI, E. Beeching, L. Tunstall, B. Lipkin, R. Soletskyi, S. C. Huang, K. Rasul, L. Yu, A. Jiang, Z. Shen, Z. Qin, B. Dong, L. Zhou, Y. Fleureau, G. Lample, and S. Polu. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

X. L. Li, A. Holtzman, D. Fried, P. Liang, J. Eisner, T. Hashimoto, L. Zettlemoyer, and M. Lewis. Contrastive decoding: Open-ended text generation as optimization, 2023. URL https://arxiv.org/abs/2210.15097.

Y. Li, L. Zheng, Y. Yu, W. Zhou, X. Zhong, X. Hu, J. Jin, H. Yuan, and T. Feng. Filter, Then Reweight: Rethinking Optimization Granularity in On-Policy Distillation, 2026a. URL http://arxiv.org/abs/2606.02684.

Y. Li, Y. Zuo, B. He, J. Zhang, C. Xiao, C. Qian, T. Yu, H.-a. Gao, W. Yang, Z. Liu, et al. Rethinking On-Policy Distillation of Large Language Models: Phenomenology, Mechanism, and Recipe, 2026b. URL https://arxiv.org/abs/2604.13016. Version Number: 2.

S. Lin, J. Hilton, and O. Evans. TruthfulQA: Measuring how models mimic human falsehoods. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Annual Meeting of the Association for Computational Linguistics*, pages 3214–3252. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2022.acl-long.229.

K. Lu and T. M. Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Y. Meng, M. Xia, and D. Chen. Simpo: Simple preference optimization with a reference-free reward, 2024. URL https://arxiv.org/abs/2405.14734.

R. Ming, H. Wu, S. Hu, Z. He, and B. Yu. One-Token Rollout: Guiding Supervised Fine-Tuning of LLMs with Policy Gradient, 2025. URL http://arxiv.org/abs/2509.26313.

S. Mukherjee, L. Yuan, D. Hakkani-Tur, and H. Peng. Reinforcement Learning Finetunes Small Subnetworks in Large Language Models, 2025. URL https://arxiv.org/abs/2505.11711v2.

S. J. Paech. Eq-bench: An emotional intelligence benchmark for large language models, 2024. URL https://arxiv.org/abs/2312.06281.

J. Pan, J. Zhang, X. Wang, L. Yuan, H. Peng, and A. Suhr. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025. Accessed: 2025-01-24.

A. Parrish, A. Chen, N. Nangia, V. Padmakumar, J. Phang, J. Thompson, P. M. Htut, and S. R. Bowman. BBQ: A hand-built bias benchmark for question answering. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Findings*, pages 2086–2105. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2022.findings-acl.165.

S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. *International Conference on Learning Representations*, 2018.

M. Rofin, A. Varre, and N. Flammarion. (How) Learning Rates Regulate Catastrophic Overtraining, 2026. URL http://arxiv.org/abs/2604.13627.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL https://arxiv.org/abs/1910.01108.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. URL http://arxiv.org/abs/2402.03300.

I. Shenfeld, J. Pari, and P. Agrawal. RL's Razor: Why Online Reinforcement Learning Forgets Less, 2025. URL https://arxiv.org/abs/2509.04259v1.

I. Shenfeld, M. Damani, J. Hübotter, and P. Agrawal. Self-Distillation Enables Continual Learning, 2026. URL http://arxiv.org/abs/2601.19897.

M. Song and M. Zheng. A survey of on-policy distillation for large language models, 2026. URL https://arxiv.org/abs/2604.00626.

Y. Tang and R. Munos. On a few pitfalls in KL divergence gradient estimation for RL, 2025. URL http://arxiv.org/abs/2506.09477.

P. Wang, L. Li, Z. Shao, R. X. Xu, D. Dai, Y. Li, D. Chen, Y.Wu, and Z. Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations, 2024a. URL https://arxiv.org/abs/2312.08935.

Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 37*, volume 37, pages 95266–95290. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024b. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad236edc564f3e3156e1b2feafb99a24-Paper-Datasets_and_Benchmarks_Track.pdf.

J. Wu, W. Deng, X. Li, S. Liu, T. Mi, Y. Peng, Z. Xu, Y. Liu, H. Cho, C.-I. Choi, et al. Medreason: Eliciting factual medical reasoning steps in llms via knowledge graphs, 2025. URL https://arxiv.org/abs/2504.00993.

X.-Y. Xiao, B. Xia, B. Yang, B. Gao, B. Shen, C. Zhang, C. He, C. Lou, F.-L. Luo, G. Wang, et al. MiMo-V2-Flash Technical Report, 2026. URL http://arxiv.org/abs/2601.02780.

W. Xu, R. Han, Z. Wang, L. T. Le, D. Madeka, L. Li, W. Y. Wang, R. Agarwal, C.-Y. Lee, and T. Pfister. Speculative Knowledge Distillation: Bridging the Teacher-Student Gap Through Interleaved Sampling, 2024. URL http://arxiv.org/abs/2410.11325.

A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024. URL https://arxiv.org/abs/2409.12122.

A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv*, 2025a.

Q. A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B.-W. Yu, C. Li, D. Liu, F. Huang, G. Dong, et al. Qwen2.5 technical report, 2025b. URL https://arxiv.org/abs/2412.15115.

G. Yu, W. Liu, Y. Hu, H.-X. Ma, J.-P. Jiang, and H.-J. Ye. Dense Supervision, Sparse Updates: On the Sparsity and Geometry of On-Policy Distillation, 2026. URL http://arxiv.org/abs/2606.13657.

Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, W. Dai, T. Fan, G. Liu, L. Liu, et al. DAPO: An Open-Source LLM Reinforcement Learning System at Scale, 2025. URL http://arxiv.org/abs/2503.14476.

L. Yuan, G. Cui, H. Wang, N. Ding, X. Wang, J. Deng, B. Shan, H. Chen, R. Xie, Y. Lin, et al. Advancing llm reasoning generalists with preference trees, 2024.

L. Yuan, W. Chen, Y. Zhang, G. Cui, H. Wang, Z. You, N. Ding, Z. Liu, M. Sun, and H. Peng. From f (x) and g (x) to f (g (x)): Llms learn new skills in rl by composing old ones. In *arXiv.org*, volume 2026, pages 147547–147574, 2026. doi: 10.48550/arXiv.2509.25123.

Y. Yue, Z. Chen, R. Lu, A. Zhao, Z. Wang, Y. Yue, S. Song, and G. Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems 38*, pages 64304–64339. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-1933.

D. Zhang, Y. Xu, H. Wang, Q. Chen, and H. Peng. Good SFT Optimizes for SFT, Better SFT Prepares for Reinforcement Learning, 2026a. URL http://arxiv.org/abs/2602.01058.

M. Zhang, Y. Liu, S. Lin, X. Yang, Q. Dai, C. Luo, W. Jiang, P. Hou, A. Zeng, X. Geng, et al. Towards On-Policy SFT: Distribution Discriminant Theory and its Applications in LLM Training, 2026b. URL http://arxiv.org/abs/2602.12222.

J. Zhou, T. Lu, S. Mishra, S. Brahma, S. Basu, Y. Luan, D. Zhou, and L. Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

S. Zhu, X. Ye, H. Lu, W. Shi, and G. Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes, 2026. URL https://arxiv.org/abs/2605.11182.

APPENDIX CONTENTS

A Additional Experimental Results **16**

A.1 Training without, rather than with, gradient clipping 16

A.2 Training with Sampled KL, Rather than Full-Vocabulary KL 18

A.3 Training on a mathematical reasoning dataset requiring longer rollouts 19

A.4 Output Coverage and Generalisation at the Higher Learning Rate 21

A.5 Additional Results on Qwen2.5-1.5B-Instruct 22

A.6 Per-benchmark analysis 24

A.7 Sparsity Analysis with a Smaller Threshold Value 24

A.8 Learning and Forgetting Curves for Science, MedReason and Countdown 25

A.9 Comparison of the Likelihoods of Student and Teacher Trajectories During OnPD and OffPD 26

A.10 A Low Forward KL Can Conceal High-Entropy On-Policy Rollouts 27

A.11 Rollout Source Affects Teacher-Style Adoption 29

A.12 Rollout-policy temperature ablation 30

B Derivation of the Token-Level KL Gradients **31**

C Rollout-Policy Sensitivity of Forward and Reverse KL Semi-Gradients **32**

C.1 Setup 32

C.2 A rollout-stability bound for forward KL 33

C.3 Reverse KL admits no distribution-free rollout-stability bound 35

C.4 A reverse-KL bound under controlled likelihood ratios 36

C.5 Relation to sequence-level KL divergences 37

D Experimental Details **39**

D.1 Datasets 39

D.2 Models 40

D.3 Training Details 41

D.4 Parameter-Update Sparsity 42

D.5 Implementation of the Rollout-Policy Spectrum 43

D.6 RLVR after Distillation 44

D.7 Spanish-Style Transfer Experiment 46

E Checkpoint Sparsity Analysis **47**

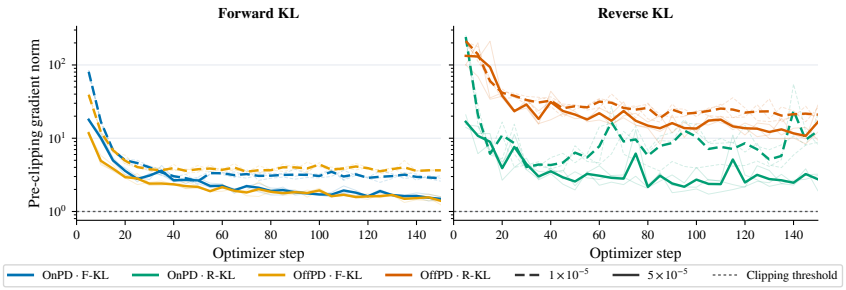


Figure 7: **Gradient clipping is active throughout Countdown-3 training.** We show the global gradient norm before clipping for the Countdown-3 runs which use gradient clipping, with thin lines denoting individual seeds and thick lines their mean across three seeds. The horizontal line marks the clipping threshold of 1.0. Every recorded norm exceeds this threshold, demonstrating that clipping rescales the gradients at every logged optimization step. Reverse KL produces substantially larger and more variable gradient norms than forward KL, including pronounced early spikes, supporting the role of clipping in stabilising reverse-KL training.

A Additional Experimental Results

A.1 Training without, rather than with, gradient clipping

Setup. our main experiments clip the global gradient norm at 1.0, and as shown in Figure 7, the global gradient norm measured before clipping exceeds our threshold of 1.0 during most training steps. To test whether the limited effect of rollout policy observed in Section 4 is caused by the use of gradient clipping, we repeat the Countdown-3 experiments with clipping disabled. For each combination of OnPD and OffPD with forward and reverse KL, we sweep learning rates from 1×10^{-5} to 5×10^{-5} while keeping all other experimental conditions fixed.

Results. The results in Figure 8 reveal a pronounced difference between forward and reverse KL when gradient clipping is removed. Forward KL remains effective at larger learning rates, particularly with OffPD, whereas reverse-KL performance deteriorates substantially. This reinforces the analysis in Section 5: the token-level reverse-KL gradient is weighted by an unbounded log-probability ratio (see Equation (5)), allowing individual tokens to produce unusually large gradients. Without clipping, these high-magnitude contributions can dominate an update and increase optimisation variance, making reverse-KL training less stable.

When it comes to catastrophic forgetting and parameter-update sparsity, here, even without the gradient clipping, the performance differences are still largely determined by the learning rate, with some extra variability coming from the rollout policy (OnPD with forward KL leads to particularly low OOD performance after training). Overall, this ablation suggests that gradient clipping is important for stabilising reverse-KL optimisation, but does not explain the generally limited effect of rollout policy observed in our main experiments.

Why does gradient clipping affect final performance, but not forgetting or update sparsity? Gradient clipping appears important for stabilising reverse KL, but does not explain the dominant effect of learning rate over rollout policy on forgetting and parameter-update sparsity. One plausible explanation lies in Adam's dynamics. Adam updates parameters using $\hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)$, where m_t and v_t are moving averages of the gradient and its coordinate-wise square (Kingma and Ba, 2014). Because this normalisation largely cancels uniform changes in gradient scale, a larger raw gradient norm does not necessarily produce a proportionally larger parameter update. An isolated unclipped gradient spike can nevertheless disrupt optimisation: it briefly biases m_t towards the direction of an atypical batch while inflating the more slowly decaying v_t, which subsequently suppresses corrective updates (Reddi et al., 2018). This may be particularly harmful for OnPD, because an early disruptive update changes the policy generating subsequent rollouts, producing poorer training trajectories, while the inflated second moment v_t inhibits recovery.

Crucially, this mechanism can impair target-task learning without increasing forgetting: if Adam normalises the initial spike and then suppresses later updates, the cumulative parameter displacement

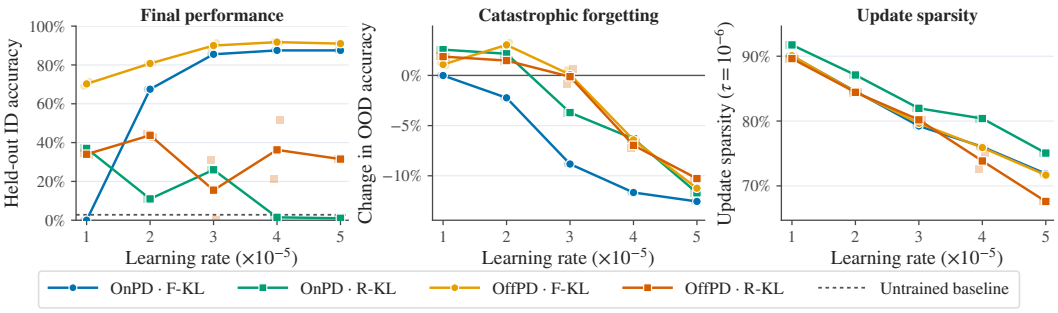


Figure 8: **The effect of learning-rate *without* gradient clipping.** Removing clipping reveals a substantial difference in final performance between forward and reverse KL, with reverse KL suffering from unstable training, while learning rate remains the dominant determinant of forgetting and parameter-update sparsity.

(including in the sensitive directions related to prior capabilities) may remain small. Consequently, gradient clipping can stabilise training, by removing the disruptive spikes in the gradient norms, while having much less influence on forgetting and parameter-update sparsity. This result is consistent with prior evidence that gradient clipping has limited influence on accumulated update sparsity in LLM post-training (Mukherjee et al., 2025).

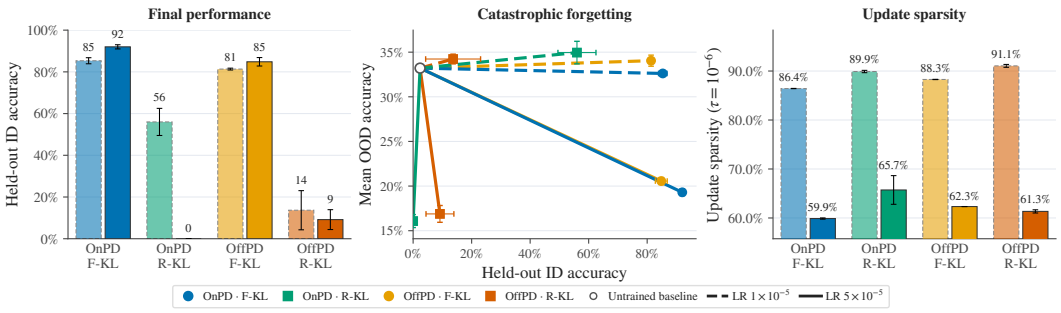


Figure 9: **Performance after training with sampled KL.** Error bars show SEMs computed over $N = 3$ seeds. Sampled-KL training exhibits the same qualitative trends as full-vocabulary KL. Forward KL achieves substantially higher and more robust final performance than reverse KL, while differences between OnPD and OffPD are comparatively limited. Catastrophic forgetting and update sparsity are governed primarily by the learning rate. These results indicate that replacing full-vocabulary KL with a single-sample gradient estimator does not materially alter our main conclusions regarding rollout policy, KL direction, and learning rate.

A.2 Training with Sampled KL, Rather than Full-Vocabulary KL

Setup. To test whether our conclusions depend on computing the KL divergence over the full vocabulary, we repeat the Countdown-3 experiments with single-sample stochastic estimators, using Llama-3.2-1B-Instruct, three seeds, learning rates 1×10^{-5} and 5×10^{-5}, and 150 training steps. Let $\pi_S^\theta(\cdot \mid x, y_{<n})$ and $\pi_T(\cdot \mid x, y_{<n})$ denote the student and teacher next-token distributions conditioned on rollout prefix $(x, y_{<n})$. OnPD constructs these prefixes from student-generated rollouts, whereas OffPD uses teacher-generated rollouts. For forward KL, the gradient is

$$\nabla_\theta D_{\mathrm{F-KL}}(\pi_S^\theta, \pi_T) = -\mathbb{E}_{y_n \sim \pi_T}\left[\nabla_\theta \log \pi_S^\theta(y_n)\right].$$

We therefore draw one token $y_n \sim \pi_T(\cdot \mid x, y_{<n})$ and use

$$\widehat{\mathcal{L}}_{\mathrm{F\text{-}KL}} = -\log \pi_S^\theta(y_n \mid x, y_{<n}).$$

The omitted teacher log-probability is constant with respect to θ, so this gives an unbiased estimator of the full forward-KL gradient. For reverse KL,

$$\nabla_\theta D_{\mathrm{R-KL}}(\pi_S^\theta, \pi_T) = \mathbb{E}_{y_n \sim \pi_S^\theta}\left[\left(\log \pi_S^\theta(y_n \mid x, y_{<n}) - \log \pi_T(y_n \mid x, y_{<n})\right) \nabla_\theta \log \pi_S^\theta(y_n \mid x, y_{<n})\right].$$

Accordingly, we draw $x_n \sim \pi_S^\theta(\cdot \mid x, y_{<n})$ and optimise the score-function surrogate

$$\widehat{\mathcal{L}}_{\mathrm{R\text{-}KL}} = \mathrm{sg}\left[\log \pi_S^\theta(y_n \mid x, y_{<n}) - \log \pi_T(y_n \mid x, y_{<n})\right] \log \pi_S^\theta(y_n \mid x, y_{<n}),$$

where sg denotes stop-gradient operation. Thus, the rollout policy determines the distribution of prefixes, while the KL direction determines the distribution from which the additional token is sampled: the teacher for forward KL and the student for reverse KL. This token is sampled independently at every valid completion position and need not coincide with the token appearing in the rollout. We average the sampled losses over all valid tokens within each completion and subsequently over the training batch, keeping the remaining training and evaluation settings matched to the full-vocabulary experiments.

Results. The results presented in Figure 9 for training with sampled KL show patterns which we have also identified when using full-vocabulary KL. Namely, we observe that OnPD and OffPD lead to very similar final performance when controlling for the KL direction and learning rate. Further, forward KL is much more robust to the rollout policy than reverse KL, further supporting the findings of Section 5. When it comes to catastrophic forgetting and update sparsity, we observe that both of these properties are primarily governed by the learning rate, similarly as when training using the full-vocabulary KL. Thus, we conclude that using the full-vocabulary KL in our main experimental results is not significantly affecting our final conclusions, and does not significantly contribute to the observed limited effects of rollout policy in the settings we study.

A.3 TRAINING ON A MATHEMATICAL REASONING DATASET REQUIRING LONGER ROLLOUTS

Setup. To test whether our conclusions extend to a more challenging reasoning domain, where finding the correct answer requires producing longer reasoning traces, we distill Qwen2.5-Math-1.5B-Instruct (Yang et al., 2024) into Qwen2.5-1.5B-Instruct using the 20K-example Numina–MATH training set described below. We consider all combinations of OnPD and OffPD rollouts with forward and reverse full-vocabulary KL, using learning rates of 1×10^{-5} and 5×10^{-5}. OnPD completions are sampled from the student, whereas OffPD completions are sampled from the teacher; in both cases, we use generation temperature 0.7, top-$p = 0.8$ (following guidelines from Yang et al. (2024)), and a maximum completion length of 1,024 tokens. The student and teacher distributions used to compute the KL objective have temperature 1. We train all model parameters for 1,000 optimization steps with an effective batch size of 128 and a constant learning rate following 10 warm-up steps. We evaluate checkpoints every 100 optimisation steps and report, for each condition, the maximum observed MATH-500 accuracy along its training trajectory. This metric is intended to compare the peak performance reached under a common training and evaluation budget, rather than to estimate the held-out performance of a checkpoint selected by an independent validation procedure. Because checkpoint selection and reporting use the same evaluations, the reported maxima may be optimistic, particularly for conditions with more variable learning curves. We therefore additionally report the complete trajectories and final-checkpoint performance. Catastrophic forgetting and update sparsity are evaluated at the corresponding selected checkpoint, and also reported over the course of training.

Numina–MATH training set. We construct a 20,000-example mathematical-reasoning dataset by combining all 7,498 problems from the training split of MATH (Hendrycks et al., 2021) with 12,502 synthetic problems sampled from the `synthetic_math` subset of NuminaMath-CoT (LI et al., 2024). For each NuminaMath example, we require a non-empty problem and solution containing a non-empty, balanced final `\boxed{}` answer. We normalize problem text by lowercasing and collapsing whitespace, remove duplicates, and exclude every synthetic problem matching either the MATH training or test split. We additionally restrict synthetic solutions to at most 384 tokens, measured using the DeepSeek-R1-Distill-Llama-8B tokenizer, and sample the remaining examples using seed 42. Responses are considered correct when their final boxed answer is symbolically equivalent to the reference answer. We use the prompt format recommended for Qwen2.5-Math. Each example contains a system message stating, "Please reason step by step, and put your final answer within `\boxed{}`," followed by the problem as the user message. We render these messages using the model's native chat template and append the assistant-generation prompt. No demonstrations or reference solutions are included. We use the same prompt format for teacher and student rollouts, KL scoring, and MATH-500 evaluation.

Inference during training. We used vLLM (Kwon et al., 2023) as a colocated inference engine for efficient rollout generation. In the OnPD setting, vLLM samples from the student policy, whose latest weights are synchronized with the training model before each rollout batch. To correct residual differences between the vLLM sampling distribution q and the policy evaluated by the training backend p_θ, following Shenfeld et al. (2026) we assign each completion the detached importance weight

$$w(y) = \frac{1}{|y|} \sum_{t=1}^{|y|} \min(2, \exp[\log p_\theta(y_t \mid x, y_{<t}) - \log q(y_t \mid x, y_{<t})]) \,.$$

This sequence-level weight is applied uniformly to the token-level distillation loss. Clipping the token ratios at 2 limits the variance of the correction, while synchronizing weights at every optimizer step keeps the generated trajectories on-policy.

Results. The results in Figure 10 largely reproduce the patterns observed in our main experiments. All conditions improve upon the untrained MATH-500 accuracy of 53.1%, with the selected checkpoints obtaining between 53.3% and 59.0%. When controlling for the KL direction and learning rate, we observe that in this regime OnPD outperforms OffPD on average, which is consistent with the finding that OnPD helps with generalisation to more difficult tasks (given that MATH-500 is a particularly challenging subset of the MATH test set). More broadly, we observe that again, forward KL seems to be more robust to the rollout policy than reverse KL, with the performance of the reverse KL being highly dependent on the learning rate. Most importantly, **the learning rate continues to govern catastrophic forgetting and update sparsity**. At 1×10^{-5}, all methods approximately preserve

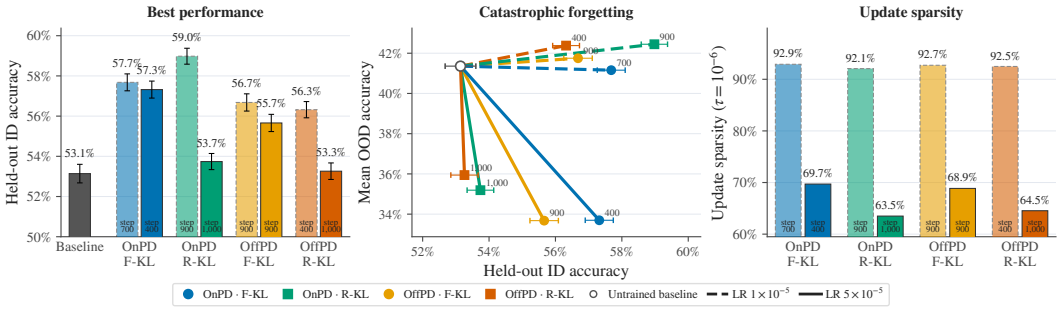


Figure 10: **Math distillation results on MATH-500.** We report the best MATH-500 performance attained during training, movement from the untrained baseline in ID–OOD space, and parameter-update sparsity at the corresponding selected checkpoint. In this regime, OnPD with reverse-KL outperforms other conditions. However, the learning rate still largely controls catastrophic forgetting and parameter-update sparsity. The performance of forward KL is also more robust to the changes in rollout policy and learning rate.

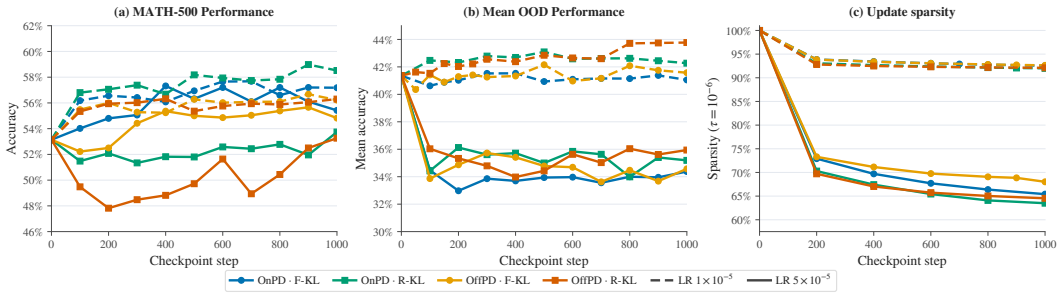


Figure 11: **MATH-500 performance throughout training.** Accuracy is evaluated every 100 optimization steps using ten generations per problem. Performance generally improves beyond the untrained baseline, although the optimal checkpoint varies considerably across conditions. The lower learning rate produces more stable learning trajectories, while the larger learning rate is particularly unstable with reverse KL. Looking at catastrophic forgetting and parameter-update sparsity, differences between OnPD and OffPD remain comparatively modest relative to the effects of learning rate and KL direction.

the baseline OOD performance while retaining 92–93% update sparsity. Increasing the learning rate to 5×10^{-5} reduces mean OOD accuracy from approximately 41.4% to 33.7–35.9% and reduces update sparsity to 63–70%. Thus, even in a longer-horizon mathematical-reasoning setting, the rollout policy has a comparatively limited effect, while the learning rate primarily determines the extent of forgetting and the density of parameter updates. We additionally provide the learning curves of the models in Figure 11, demonstrating how the MATH-500 performance, OOD performance, and update sparsity vary over the course of training. Just as in the case of other datasets (cf. Figure 19), the majority of the changes to the OOD performance happen in the initial steps of training.

A.4 Output Coverage and Generalisation at the Higher Learning Rate

Figure 12 complements Figure 5 with learning rate 5×10^{-5}. Both figures use checkpoint 150 and matched distillation seeds 42, 43, and 44. Each evaluation uses the first 200 test questions, ten sampled responses per question, and evaluation seed 42. Countdown-3 uses temperature 1.0; Countdown-4E uses temperature 0.5 and a 4,096-token completion limit. Countdown-4 has four operands instead of three which increases the search space significantly, with operands drawn from 1–10. Shading denotes the pass@1-to-pass@10 span, not uncertainty; error bars denote SEM across distillation seeds. Under the larger learning rate, the performance generally decreases significantly, particularly when training with Reverse KL. Hence, we consider the trends observed in these figures less informative than those for learning rate 1×10^{-5}.

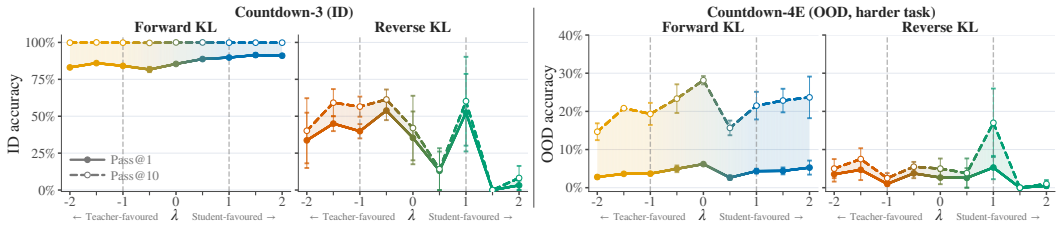


Figure 12: **Output coverage and generalisation at LR 5×10^{-5} for Llama3.2-1B.** Same protocol and plotting conventions as Figure 5; means and SEM over three distillation checkpoints.

A.5 Additional Results on Qwen2.5-1.5B-Instruct

To assess whether our findings generalise beyond the Llama 3 family, we repeat the main experiments using Qwen2.5-1.5B-Instruct as the student and Qwen2.5-7B-Instruct as the teacher (Yang et al., 2025b). We strengthen the teacher using the same training pipeline described in Appendix D.2.2. We use Qwen2.5 rather than Qwen3 to avoid potential confounding effects from Qwen3's explicit reasoning training which can be activated in the "thinking mode".

Performance and forgetting. Figure 13 reports final held-out performance across the three datasets, while Figure 14a shows the corresponding changes in OOD performance. The results are consistent with those obtained for Llama3 in the main text. We find no consistent performance advantage for either OnPD or OffPD across tasks and learning rates. Instead, KL direction has a clearer effect on target-task performance, with forward KL generally outperforming reverse KL, while learning rate remains the primary determinant of catastrophic forgetting. These results suggest that our main conclusions are robust across the two model families.

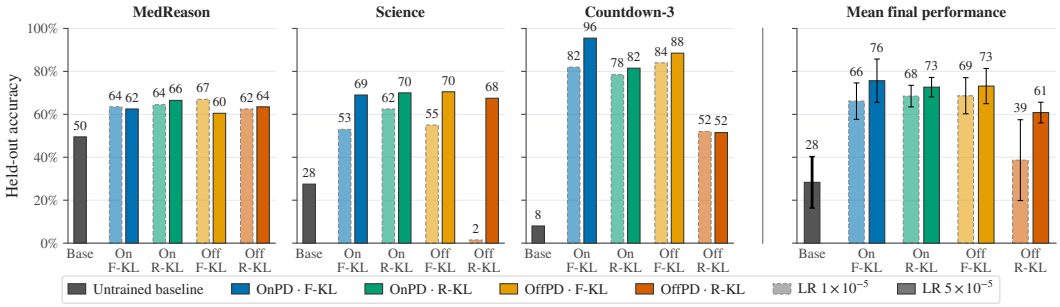
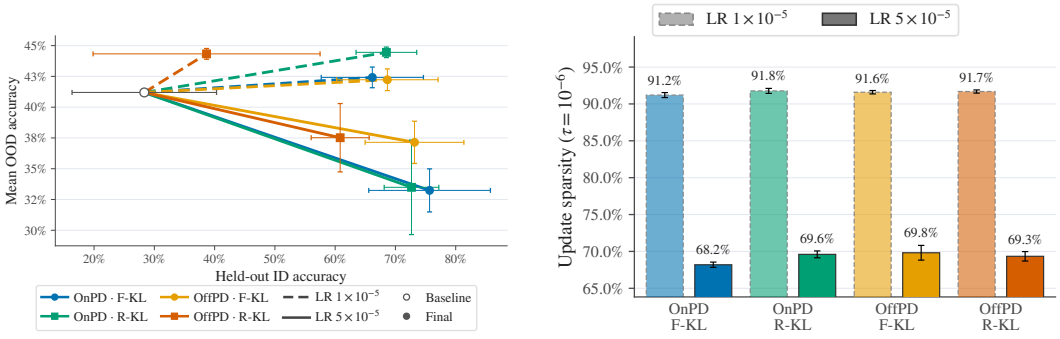


Figure 13: **Comparison of the effectiveness of OnPD and OffPD training for Qwen2.5-1.5B.** Dataset panels report mean held-out accuracy over $N = 3$ repeats; error bars show SEM. The rightmost panel shows the average across the nine runs. Across the evaluated settings, KL direction appears to have a larger effect on final performance than rollout policy, with forward KL outperforming reverse KL on average.



(a) Catastrophic Forgetting

(b) Parameter-Update Sparsity

Figure 14: **Catastrophic forgetting and parameter-update sparsity for Qwen2.5-1.5B.** Similarly as for Llama3.2-1B, we observe that for Qwen2.5-1.5B learning rate is a stronger predictor of the degree of catastrophic forgetting and the parameter-update sparsity than the rollout policy. *Left:* The mean OOD accuracy consistently decreases over the course of training when the larger learning rate of 5×10^{-5} is used, but increases when the smaller learning rate is used instead. OffPD does not lead to stronger forgetting in any of the conditions. *Right:* Similarly, parameter-update sparsity does not vary significantly between the rollout policies, but depends strongly on the learning rate. For both figures, error bars mark SEM computed over the three datasets.

Sensitivity to rollout policy. Further, we repeat the rollout-spectrum experiment from Figure 3 using Qwen2.5-1.5B. As shown in Figure 15, forward-KL remains comparatively robust across the

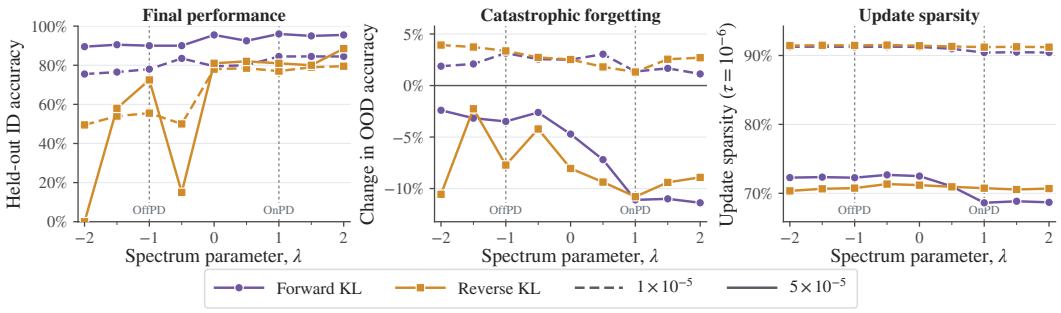


Figure 15: **Sensitivity of Qwen2.5-1.5B to the rollout policy.** Results at checkpoint 150 across the contrastive-decoding spectrum, using seed 42. The performance of forward-KL is robust to the rollout policy, while the performance of reverse-KL degrades sharply under teacher-favoured rollouts. Consistent with previous findings, larger learning rates amplify forgetting, particularly for student-favoured rollouts, while also producing denser parameter updates.

spectrum, whereas reverse-KL is highly sensitive to teacher-favoured rollouts ($\lambda < 0$), with several settings exhibiting severe performance degradation. For $\lambda \geq 0$, reverse-KL recovers substantially and approaches forward-KL performance. The learning rate primarily controls forgetting and update sparsity. At 5×10^{-5}, updates are markedly less sparse and cause substantially more forgetting than at 10^{-5}. Surprisingly, within this high-learning-rate regime, student-favoured rollouts ($\lambda > 0$) produce increasingly severe forgetting under forward-KL and generally greater forgetting under reverse-KL, although the latter is less monotonic. Thus, the Qwen results reproduce the main qualitative asymmetry observed with Llama-3.2, while providing additional evidence showing that on-policy rollouts *need not* reduce catastrophic forgetting.

Output coverage and generalisation along the rollout-policy spectrum. Figure 16 repeats the Countdown-3/Countdown-4E comparison in Figure 5 for Qwen2.5-1.5B-Instruct at both learning rates. The evaluation protocol is the same as in Appendix A.4, using the distillation checkpoint from seed 42.

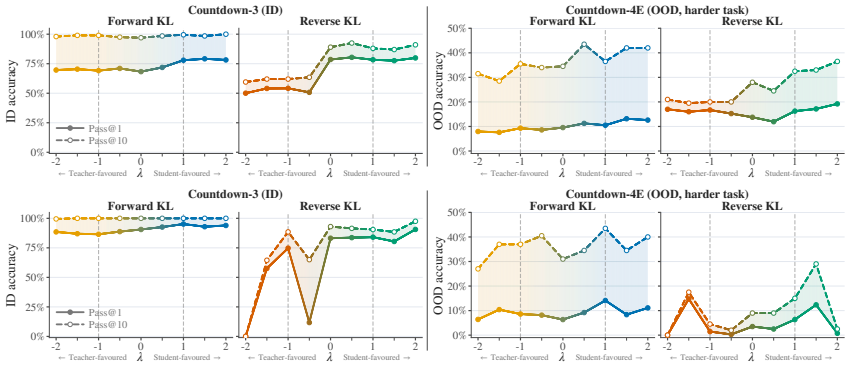


Figure 16: **Output coverage and generalisation for Qwen2.5-1.5B.** LR 1×10^{-5} (top) and 5×10^{-5} (bottom). Same plotting conventions as Figure 5.

A.6 PER-BENCHMARK ANALYSIS

Figure 17 shows the accuracy across all 7 OOD datasets. Higher learning rate leads to strong losses on MMLU-Pro, HumanEval-Instruct, IFEval, and EQ-Bench under both rollout policies. TruthfulQA and BBQ improve slightly for most configurations. Within individual benchmarks, differences between OnPD and OffPD are generally smaller than the learning-rate effect and do not have a consistent direction. The three training-task markers show that this conclusion is also largely stable across the domain on which distillation is performed.

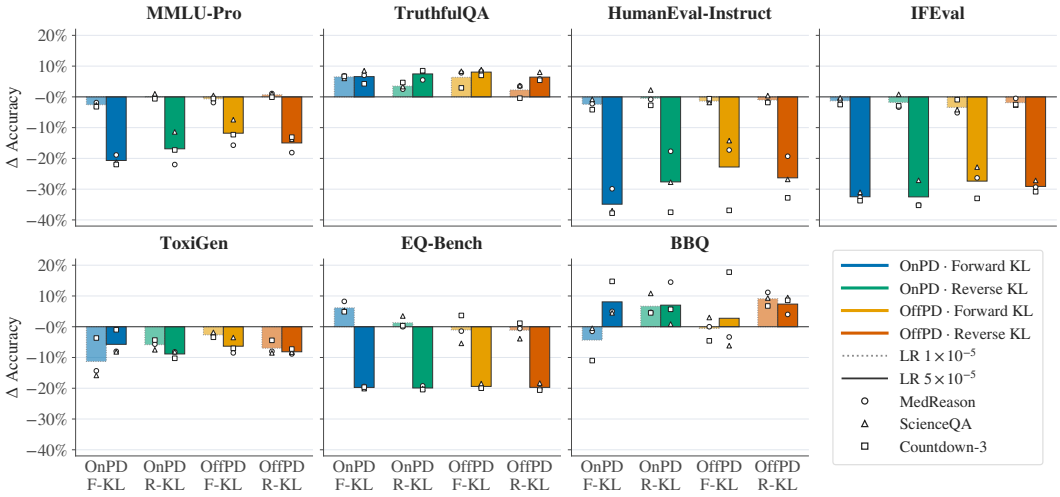


Figure 17: **Per-benchmark changes in out-of-distribution performance.** Each panel reports the change from the untrained student baseline to the final distilled checkpoint on one OOD benchmark. Bars average repeated runs within each training task and then weight MedReason, Science, and Countdown-3 equally. White markers show the corresponding training-task means.

A.7 SPARSITY ANALYSIS WITH A SMALLER THRESHOLD VALUE

To show the robustness of our results evaluating the sparsity of the parameter updates, we also report the results obtained with a more conservative threshold, $\tau = 10^{-8}$ (compared to $\tau = 10^{-6}$ reported in the main text). Figure 18 shows that using a smaller threshold does not substantially affect the results, and particularly their qualitative interpretation.

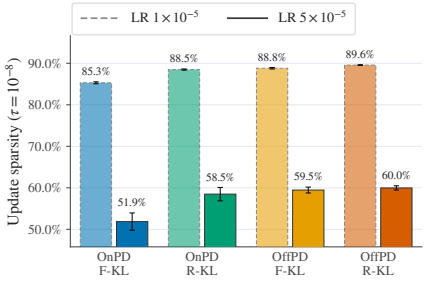


Figure 18: **Parameter-update sparsity with a smaller threshold.** Each bar pools the three datasets and $N = 3$ repeats per dataset; error bars show SEM across the nine runs. Learning rate, more than rollout policy, determines sparsity of updates.

A.8 Learning and Forgetting Curves for Science, MedReason and Countdown

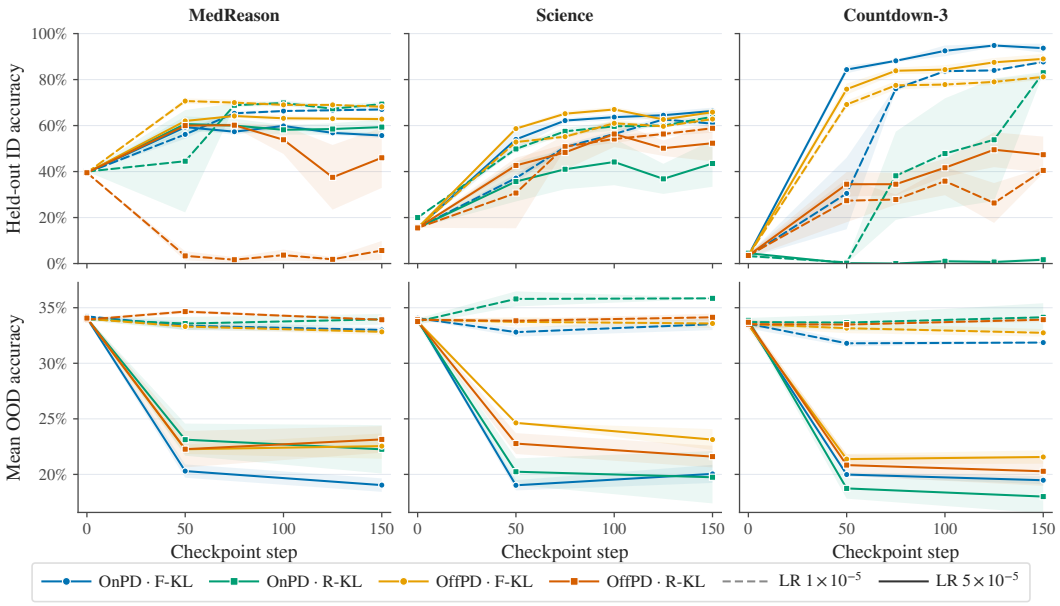


Figure 19: **Learning and forgetting curves for the main runs on Llama3.2-1B-Instruct.** We report the accuracy on the held-out test set of the training task (top row) and the mean accuracy over the out-of-distribution tasks (bottom row) over the course of training. We can observe that for the majority of the runs, the target task performance converges to a stable value after 150 optimisation steps. Additionally, most of the forgetting happens within the first 50 steps of training, with the OOD performance maintaining a relatively stable value at steps 50-150. The shading represents mean $\pm$ one standard error of the mean (SEM) computed over three seeds.

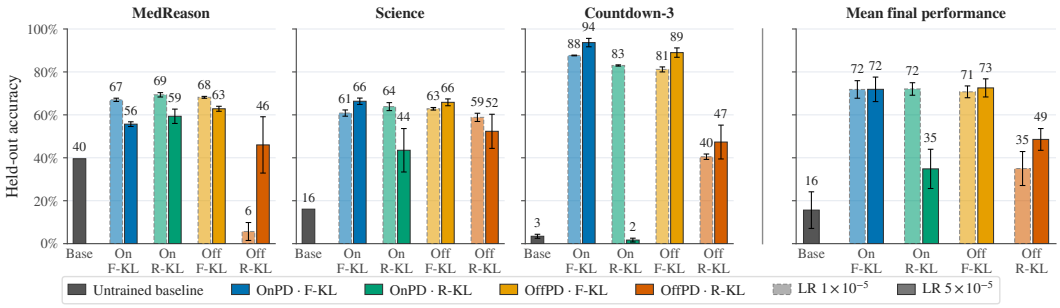


Figure 20: **Breakdown of the final performance obtained on each dataset.** We demonstrate a breakdown of the final performance after 150 steps of training, across the three datasets. Error bars denote SEM computed over $N = 3$ seeds.

A.9 Comparison of the Likelihoods of Student and Teacher Trajectories During OnPD and OffPD

In Figure 21 we visualise the likelihoods of the teacher and the student over the trajectories generated over the course of training, to validate to what extent the small visible differences between OnPD and OffPD stem from the small differences in the trajectories generated by the student and the teacher. The figure demonstrates that across the datasets, we observe significant differences between the likelihood dynamics of OnPD and OffPD.

During OffPD training runs, the teacher models are very confident on the (teacher-) generated trajectories, with average confidence $\approx 90\%$. The student confidence on the teacher-generated trajectories slowly increases over the course of training, following a stable path. However, most of the time it remains below the confidence of the teacher.

For OnPD runs, where the trajectories are generated by the student, the situation looks different. Both when using F-KL and R-KL, the student is initially confident on the trajectories (average likelihood $\approx 80\%$), while the teacher is not (average likelihood $\approx 20\%$). When training with R-Kl, both the likelihood under the teacher and under the student tends to increase during training, suggesting that the student becomes more confident in its predictions, and the quality of the generation increases (as indicated by the increasing likelihood under the teacher). However, under F-KL the situation looks different: the confidence of the student drops drastically in the initial steps, and then slowly recovers over the course of training. The average likelihood of the teacher on the student-generated traces remains relatively low throughout training, suggesting that the quality of the supervision might be low. We further analyse these peculiar dynamics of the OnPD training with F-KL in Appendix A.10.

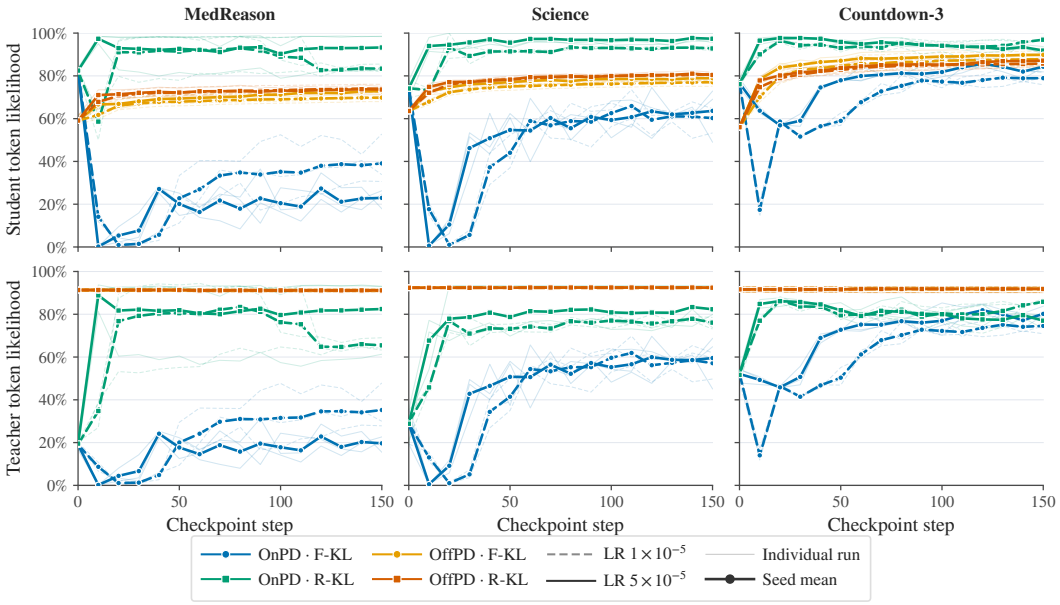


Figure 21: **Teacher and student likelihoods on the trajectories generated during OnPD and OffPD training.** Both the teacher and student likelihoods were computed at temperature 1.0.

A.10 A Low Forward KL Can Conceal High-Entropy On-Policy Rollouts

Forward KL can be expressed as:

$$D_{\mathrm{KL}}(T\|S) = \mathrm{CE}(T,S) - H(T), \tag{6}$$

where T and S denote the teacher and student next-token distributions on the sampled prefix. Consequently, a small forward KL can conceal both high cross-entropy and teacher entropy. We observed a particularly clear instance of this effect when distilling `Llama-3.2-1B-Instruct` on student-generated trajectories.

Setup. We analyze OnPD with the forward-KL objective on Science and MedReason, using dataset-specific trained Llama-3.1-8B teachers. The student is fully fine-tuned with a learning rate of 1×10^{-5}, an effective batch size of 32, student sampling temperature 1.0 and teacher distribution temperature 0.5. Every 10 steps, we evaluate token-level $D_{\mathrm{KL}}(T\|S)$, $H(T)$, and $\mathrm{CE}(T,S)$ on freshly sampled student trajectories. Metrics are binned in intervals of 50 completion tokens and reported separately for each dataset. Curves show the mean and range over 3 seeds at token positions reached by all three runs.

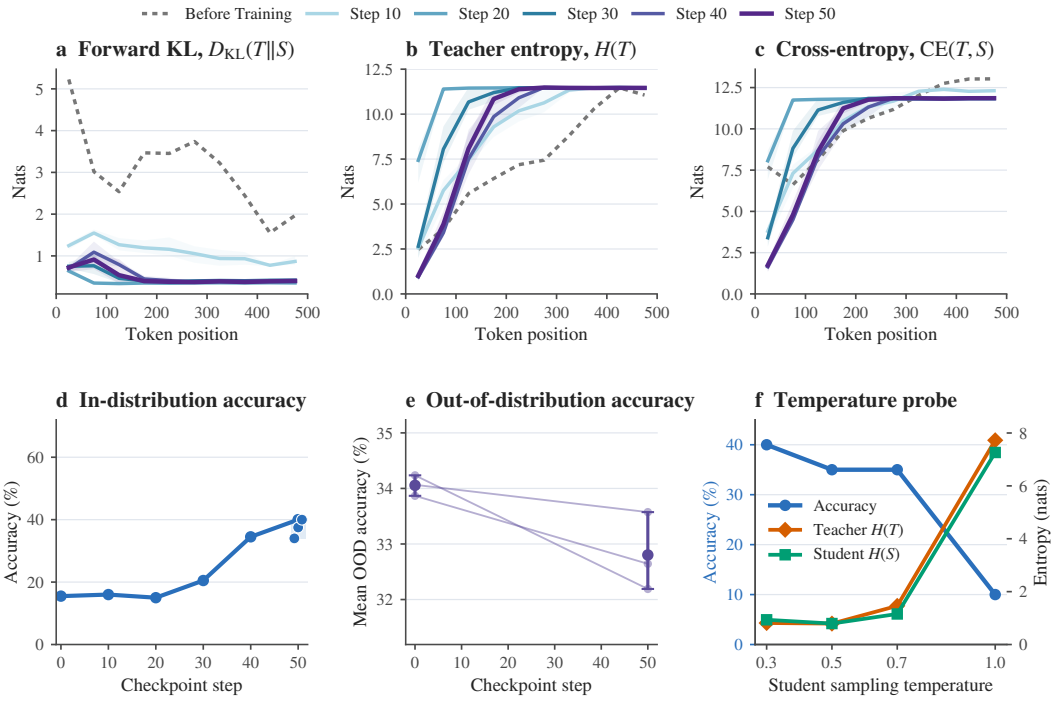


Figure 22: **Science OnPD forward-KL dynamics.** (a)–(c): token-position profiles at initialization (dotted grey) and every 10 training steps through step 50 (solid lines: three-seed means; shading: min–max ranges). (d): held-out accuracy for seed 42, with all three repeats at step 50. (e): OOD accuracy for individual seeds, with mean markers and min–max bars at initialization and step 50. (f): accuracy and teacher/student entropy when re-sampling the seed-42 checkpoint-50 model on the same 20 prompts at different temperatures.

Low KL masks high-entropy rollouts. Figure 22 shows that, during steps 10–20, forward KL becomes small while teacher entropy and cross-entropy approach the vocabulary scale (12 nats≈163k effective token choices). Since $D_{\mathrm{KL}}(T\|S) = \mathrm{CE}(T,S) - H(T)$, low KL here reflects the near-cancellation of two large terms rather than confident predictions. Where teacher and student are nearly uniform and closely matched, the distillation gradient is weak, so the useful supervision signal is concentrated at earlier token positions. As training progresses, the low-entropy region advances along the trajectory. This suggests a self-reinforcing mechanism: improvements to early tokens produce prefixes on which the teacher can provide informative supervision further into the completion,

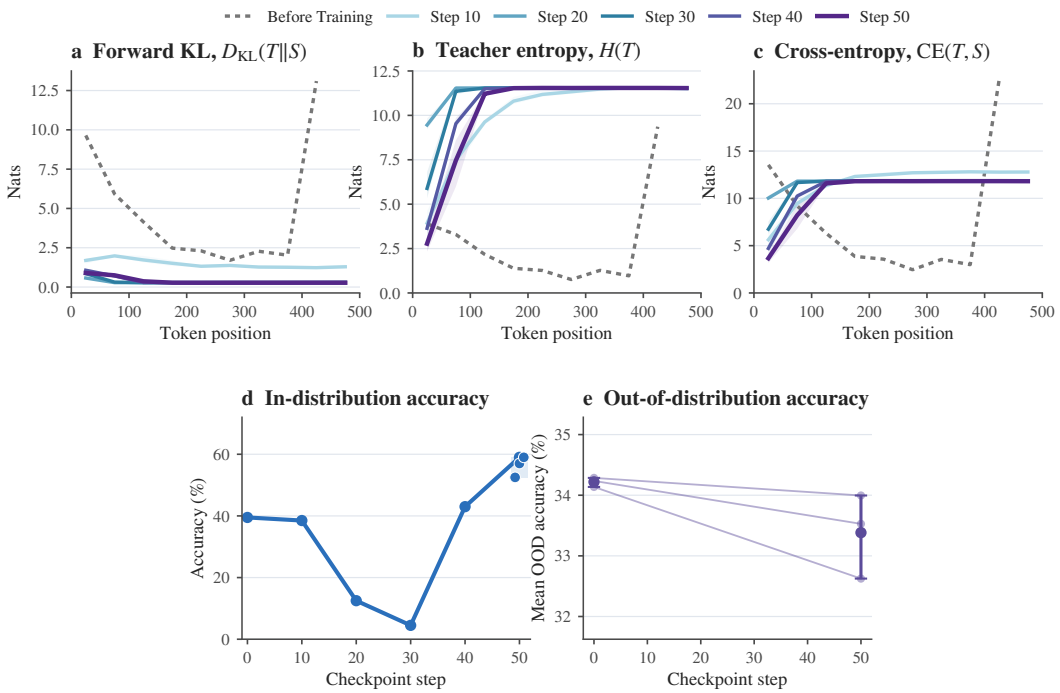


Figure 23: **MedReason OnPD forward-KL dynamics.** (a)–(c): token-position profiles of forward KL, teacher entropy, and cross-entropy. (d)–(e): held-out MedReason and OOD accuracy. Plotting conventions match Figure 22.

enabling subsequent positions to improve in turn. This particularly acute example illustrates how on-policy learning may operate when student and teacher distributions have limited overlap: by progressively extending the student's access to useful teacher supervision and steering its trajectories to remain within teacher-supported regions for longer.

Learning and temperature sensitivity. Despite these high-entropy rollouts, step-50 target performs surprisingly well, averaging 37% accuracy on Science and 56% on MedReason, with OOD declines of only 1.3 and 0.8 percentage points. We hypothesize that this discrepancy reflects differences in the sampling temperature (evaluation uses $T = 0.5$, whereas training use $T = 1.0$). To measure this effect, we re-sample the Science checkpoint-50 model on the same 20 prompts at temperatures $\{0.3, 0.5, 0.7, 1.0\}$. Accuracy remains high up to temperature 0.7 but drops sharply at temperature 1.0, where both teacher and student entropy exceed 7 nats. This suggests that rollout instability does not necessarily imply a loss of capability, but rather a failure to reliably access that capability when sampling at temperature 1.0.

A.11 Rollout Source Affects Teacher-Style Adoption

An important setting in which rollout policy may matter, even when final task performance is similar, is the transfer of incidental teacher behaviours. We therefore test whether rollout policy affects the adoption of a teacher's stylistic preferences under different distillation objectives.

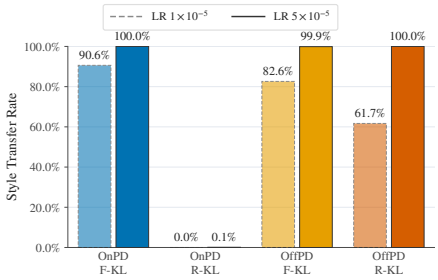


Figure 24: **Style transfer from teacher to student model.** OnPD with R-KL does not learn to generate traces in Spanish.

Experimental Setup. We train Qwen2.5-1.5B student models on Science, using as teacher a Qwen2.5-3B model conditioned on a correct demonstration (similar to self-distillation setup (Shenfeld et al., 2026)), as well as an instruction to only speak in Spanish. We then evaluate the propensity of the student model to generate responses in Spanish, using GPT-5 to annotate student-generated trajectories at test time. Full training, evaluation, and annotation details are provided in Appendix D.7.

Results Figure 24 shows the results across rollout policies, KL objectives and learning rates. Teacher style transfers to the student for most setups, except OnPD with reverse KL. The mode-seeking behaviour of reverse KL makes it more robust to style transfer than forward KL: style tokens favoured by the teacher but not by the student contribute little to the reverse KL loss, but have a large effect under forward KL. However, off-policy rollouts still lead to style transfer from the teacher to the student, even under reverse KL. This suggests that training on states produced by the teacher can bias the student towards the teacher's style even when using the more robust reverse KL objective.

The robustness of OnPD with forward KL to adopting the teacher style likely depends on the ability of the teacher to provide meaningful supervision on student-generated trajectories. In this setting, the teacher still understands English, so when it is queried on English student-generated prefixes, its next-token distribution can retain an English mode. Reverse KL can therefore match a teacher-supported continuation in terms of final performance, without forcing the student into the incidental Spanish style.

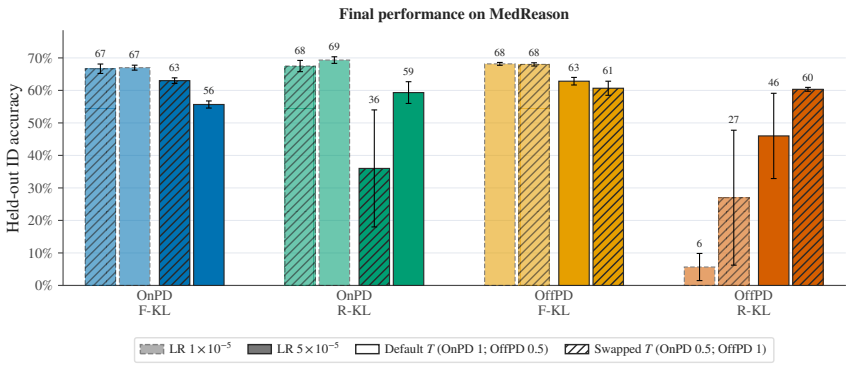


Figure 25: **Generation-temperature ablation on MedReason.** Modifying the generation temperature does not affect the best performance achieved by either OnPD or OffPD. Error bars mark the SEM over $N = 3$ seeds.

A.12 Rollout-policy temperature ablation

Setup. In our main experiments we consistently rescale our trained teacher model with temperature 0.5, affecting both the the rollout generation temperature and the temperature used in the KL objective. As we explain in Appendix D.2.2, this is to avoid degenerate generations which are occasionally produced at temperature 1.0. For the student, we consistently use temperature 1.0. In this appendix, we ablate whether using different generation temperatures for the student and the teacher could have significantly affected our results.

We repeat the results from Figure 1 on the MedReason dataset, where the teacher had a tendency to produce particularly degenerate generations at temperature 1.0 (probably due to the RLVR training). We use Llama-3.2-1B-Instruct student and the dataset-specific trained Llama-3.1-8B-Instruct teacher. We consider OnPD and OffPD, forward and reverse KL, learning rates $\{1 \times 10^{-5}, 5 \times 10^{-5}\}$, and seeds 42, 43, and 44. The original configuration samples OnPD trajectories from the student at temperature $T_{\text{gen}} = 1.0$ and OffPD trajectories from the teacher at $T_{\text{gen}} = 0.5$. In the ablation, we swap only these sampling temperatures: OnPD now uses $T_{\text{gen}} = 0.5$, while OffPD now uses $T_{\text{gen}} = 1.0$. Crucially, the temperatures used to compute the KL objective are unchanged in every condition: the student distribution is scored at $T_S = 1.0$ and the teacher distribution at $T_T = 0.5$. This isolates the effect of the trajectory-generating distribution from the effect of temperature on the training objective. We report held-out MedReason accuracy at step 150. Each bar in Figure 25 is the mean over three seeds, and error bars denote the standard error of the mean.

Results. Forward-KL performance is largely stable under the temperature swap. At the lower learning rate, OnPD changes from 67.0% to 66.7% and OffPD from 68.2% to 68.0%. At the higher learning rate, lowering the OnPD generation temperature improves accuracy from 55.7% to 63.0%, whereas raising the OffPD generation temperature reduces accuracy from 62.8% to 60.7%. Thus, the low-learning-rate forward-KL comparison is not explained by the different default sampling temperatures, although generation temperature has a moderate effect on OnPD at the higher learning rate.

Reverse KL is more sensitive to the generation distribution. For OnPD, the temperature swap changes accuracy from 69.3% to 67.5% at 1×10^{-5}, but from 59.3% to 36.0% at 5×10^{-5}; the latter condition also has high seed-to-seed variability. For OffPD, raising the generation temperature changes accuracy from 5.7% to 27.0% at 1×10^{-5} and from 46.0% to 60.3% at 5×10^{-5}. The low-learning-rate OffPD comparison is itself highly variable across seeds. Overall, the ablation indicates that the best achievable performance of OnPD and OffPD is not affected by the generation temperature, thus supporting our existing analysis.

The results in Section 5 support these conclusions further, where in creating the rollout-policy spectrum we are mixing the student and teacher rollout policies when both are scaled with temperature 1.0. Hence, at $\lambda = 1$ and $\lambda = -1$ we recover a comparison with OnPD and OffPD where both generation temperatures are set to 1.0.

B Derivation of the Token-Level KL Gradients

We derive the gradients at a fixed prefix, treating the teacher distribution as independent of θ. The forward and reverse KL objectives are

$$D_{\mathrm{F-KL}} = \sum_{u \in \mathcal{V}} \pi_T(u) \log \frac{\pi_T(u)}{\pi_S^\theta(u)}, \tag{7}$$

$$D_{\mathrm{R-KL}} = \sum_{u \in \mathcal{V}} \pi_S^\theta(u) \log \frac{\pi_S^\theta(u)}{\pi_T(u)}. \tag{8}$$

For the student softmax,

$$\frac{\partial \pi_S^\theta(u)}{\partial (z_S^\theta)_v} = \pi_S^\theta(u) \left(\mathbb{I}[u = v] - \pi_S^\theta(v) \right), \tag{9}$$

$$\frac{\partial \log \pi_S^\theta(u)}{\partial (z_S^\theta)_v} = \mathbb{I}[u = v] - \pi_S^\theta(v). \tag{10}$$

Forward KL. Differentiating the full vocabulary sum gives

$$\frac{\partial D_{\mathrm{F-KL}}}{\partial (z_S^\theta)_v} = -\sum_{u \in \mathcal{V}} \pi_T(u) \left(\mathbb{I}[u = v] - \pi_S^\theta(v) \right) \tag{11}$$

$$= -\pi_T(v) + \pi_S^\theta(v) \sum_{u \in \mathcal{V}} \pi_T(u) \tag{12}$$

$$= \pi_S^\theta(v) - \pi_T(v). \tag{13}$$

Applying the chain rule therefore yields

$$\nabla_\theta D_{\mathrm{F-KL}} = \sum_{v \in \mathcal{V}} \left(\pi_S^\theta(v) - \pi_T(v) \right) \nabla_\theta (z_S^\theta)_v.$$

Reverse KL. Let

$$r(u) = \log \frac{\pi_S^\theta(u)}{\pi_T(u)}.$$

Using $\partial [x \log x] / \partial x = \log x + 1$, we obtain

$$\frac{\partial D_{\mathrm{R-KL}}}{\partial (z_S^\theta)_v} = \sum_{u \in \mathcal{V}} \frac{\partial \pi_S^\theta(u)}{\partial (z_S^\theta)_v} \left(r(u) + 1 \right) \tag{14}$$

$$= \pi_S^\theta(v) \left(r(v) + 1 \right) - \pi_S^\theta(v) \sum_{u \in \mathcal{V}} \pi_S^\theta(u) \left(r(u) + 1 \right). \tag{15}$$

Because

$$\sum_{u \in \mathcal{V}} \pi_S^\theta(u) r(u) = D_{\mathrm{R-KL}} \qquad \text{and} \qquad \sum_{u \in \mathcal{V}} \pi_S^\theta(u) = 1,$$

this simplifies to

$$\frac{\partial D_{\mathrm{R-KL}}}{\partial (z_S^\theta)_v} = \pi_S^\theta(v) \left[\log \frac{\pi_S^\theta(v)}{\pi_T(v)} - D_{\mathrm{R-KL}} \right].$$

The parameter gradient is consequently

$$\nabla_\theta D_{\mathrm{R-KL}} = \sum_{v \in \mathcal{V}} \pi_S^\theta(v) \left[\log \frac{\pi_S^\theta(v)}{\pi_T(v)} - D_{\mathrm{R-KL}} \right] \nabla_\theta (z_S^\theta)_v.$$

This expression assumes that $\pi_T(v) > 0$ wherever $\pi_S^\theta(v) > 0$; otherwise, the reverse KL is infinite.

C Rollout-Policy Sensitivity of Forward and Reverse KL Semi-Gradients

This appendix formalizes a stability distinction between forward- and reverse-KL distillation. The relevant object is the update used by the algorithm, which stops gradients through sampled trajectories, rather than the total derivative of an on-policy objective. We show that the forward-KL semi-gradient is uniformly Lipschitz in the rollout-induced distribution over prefixes. Reverse KL has no analogous distribution-free guarantee: its semi-gradient can be arbitrarily sensitive to an arbitrarily small change in the prefix distribution. A bound for reverse KL is recovered only after controlling student–teacher log-likelihood ratios.

C.1 Setup

We retain the notation introduced in Section 3. Let ρ denote the autoregressive rollout policy. For a prompt x and completion $y = (y_1, \dots, y_{L_y})$, let $y_{<n} = (y_1, \dots, y_{n-1})$ denote the output prefix preceding token y_n, and write

$$h_n := (x, y_{<n}). \tag{16}$$

The rollout policy induces the distribution

$$\rho(y \mid x) = \prod_{n=1}^{L_y} \rho(y_n \mid h_n) \tag{17}$$

over complete output sequences.

For $D \in \{D_{\text{F-KL}}, D_{\text{R-KL}}\}$, define the rollout-conditioned distillation objective

$$\mathcal{L}_D(\theta; \rho) := \mathbb{E}_{x \sim p_{\text{data}}} \mathbb{E}_{y \sim \rho(\cdot|x)} \left[\frac{1}{L_y} \sum_{n=1}^{L_y} D\left(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\right) \right]. \tag{18}$$

Off-policy distillation corresponds to $\rho = \pi_T$, whereas on-policy distillation corresponds to $\rho = \pi_S^\theta$.

When $\rho = \pi_S^\theta$, the objective depends on θ both through the student distribution in the token-level divergence and through the sampled completion. As in our training procedure, we stop gradients through the sampling process and treat y, including every prefix h_n, as fixed. We denote the resulting semi-gradient by

$$\bar{\nabla}_\theta \mathcal{L}_D(\theta; \rho) := \mathbb{E}_{x \sim p_{\text{data}}} \mathbb{E}_{y \sim \rho(\cdot|x)} \left[\frac{1}{L_y} \sum_{n=1}^{L_y} \nabla_\theta D\left(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\right) \right], \tag{19}$$

where the distributions inside the expectations are held fixed during differentiation. If ρ does not depend on θ, this semi-gradient is the ordinary gradient of $\mathcal{L}_D(\theta; \rho)$.

Let $z_S^\theta(h_n) \in \mathbb{R}^{|\mathcal{V}|}$ denote the student logits at prefix h_n, so that

$$\pi_S^\theta(\cdot \mid h_n) = \mathrm{softmax}\big(z_S^\theta(h_n)\big), \tag{20}$$

and define the corresponding logit Jacobian by

$$J_\theta(h_n) := \frac{\partial z_S^\theta(h_n)}{\partial \theta}. \tag{21}$$

The parameter gradient of either token-level divergence can therefore be written as

$$\begin{aligned} &\nabla_\theta D\left(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\right) \\ &\quad = J_\theta(h_n)^\top \nabla_{z_S^\theta(h_n)} D\left(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\right). \end{aligned} \tag{22}$$

C.2 A ROLLOUT-STABILITY BOUND FOR FORWARD KL

At any fixed prefix, the forward-KL gradient with respect to the student logits is

$$\nabla_{z_S^\theta(h_n)} D_{\text{F-KL}}\left(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\right) = \pi_S^\theta(\cdot \mid h_n) - \pi_T(\cdot \mid h_n). \tag{23}$$

For conciseness, define the forward-KL parameter-gradient contribution of a complete sampled trajectory as

$$g^\theta_{\text{F-KL}}(x, y) := \frac{1}{L_y} \sum_{n=1}^{L_y} J_\theta(h_n)^\top \Big(\pi_S^\theta(\cdot \mid h_n) - \pi_T(\cdot \mid h_n)\Big). \tag{24}$$

Equation 19 then gives

$$\bar{\nabla}_\theta \mathcal{L}_{D_{\text{F-KL}}}(\theta; \rho) = \mathbb{E}_{x \sim p_{\text{data}}} \mathbb{E}_{y \sim \rho(\cdot|x)} \left[g^\theta_{\text{F-KL}}(x, y)\right]. \tag{25}$$

We use $\text{TV}(P, Q) = \frac{1}{2}\|P - Q\|_1$. When applied to $\rho(\cdot \mid x)$ and $\rho'(\cdot \mid x)$, this is the total-variation distance between the distributions over complete output sequences induced by the two rollout policies for the same prompt x.

Theorem C.1 (Forward-KL rollout stability)**.** *Suppose that there exists $B > 0$ such that, for every prompt x, every completion y having positive probability under $\rho(\cdot \mid x)$ or $\rho'(\cdot \mid x)$, every token position n, and every vector $v \in \mathbb{R}^{|\mathcal{V}|}$,*

$$\left\|J_\theta(h_n)^\top v\right\|_2 \le B\|v\|_2. \tag{26}$$

Equivalently, B uniformly bounds the largest singular value of the student-logit Jacobian over all prefixes encountered under either rollout policy. Then

$$\left\|\bar{\nabla}_\theta \mathcal{L}_{D_{\text{F-KL}}}(\theta; \rho) - \bar{\nabla}_\theta \mathcal{L}_{D_{\text{F-KL}}}(\theta; \rho')\right\|_2 \le 2\sqrt{2}B\, \mathbb{E}_{x \sim p_{\text{data}}} \left[\text{TV}(\rho(\cdot \mid x), \rho'(\cdot \mid x))\right]. \tag{27}$$

Moreover, for every rollout policy ρ,

$$\mathbb{E}_{x \sim p_{\text{data}}} \mathbb{E}_{y \sim \rho(\cdot|x)} \left[\left\|g^\theta_{\text{F-KL}}(x, y)\right\|_2^2\right] \le 2B^2. \tag{28}$$

Proof. We first bound the forward-KL gradient at one prefix. For any probability vectors p and q, $\|p - q\|_2 \le \sqrt{2}$. Applying Equation 26 with

$$v = \pi_S^\theta(\cdot \mid h_n) - \pi_T(\cdot \mid h_n) \tag{29}$$

and using Equation 23 gives the per-token bound

$$\left\|J_\theta(h_n)^\top \Big(\pi_S^\theta(\cdot \mid h_n) - \pi_T(\cdot \mid h_n)\Big)\right\|_2 \le B \left\|\pi_S^\theta(\cdot \mid h_n) - \pi_T(\cdot \mid h_n)\right\|_2 \le \sqrt{2}B. \tag{30}$$

We next pass from a single token to a complete sampled trajectory. Substituting Equation 24 and applying the triangle inequality to the sum over token positions yields

$$\left\|g^\theta_{\text{F-KL}}(x, y)\right\|_2 = \left\|\frac{1}{L_y} \sum_{n=1}^{L_y} J_\theta(h_n)^\top \Big(\pi_S^\theta(\cdot \mid h_n) - \pi_T(\cdot \mid h_n)\Big)\right\|_2 \tag{31}$$

$$\le \frac{1}{L_y} \sum_{n=1}^{L_y} \left\|J_\theta(h_n)^\top \Big(\pi_S^\theta(\cdot \mid h_n) - \pi_T(\cdot \mid h_n)\Big)\right\|_2 \tag{32}$$

$$\le \frac{1}{L_y} \sum_{n=1}^{L_y} \sqrt{2}B = \sqrt{2}B. \tag{33}$$

This holds pointwise for every x and y.

We now compare the expectations of this trajectory-level gradient under two rollout policies. For any bounded vector-valued function g and discrete probability distributions P and Q,

$$\|\mathbb{E}_P[g] - \mathbb{E}_Q[g]\|_2 \leq 2 \sup_y \|g(y)\|_2 \, \mathrm{TV}(P, Q). \tag{34}$$

Applying Equation 34 conditionally for each prompt, with $P = \rho(\cdot \mid x)$, $Q = \rho'(\cdot \mid x)$, and $g(y) = g^{\theta}_{\text{F-KL}}(x, y)$, yields

$$\begin{aligned} &\left\| \mathbb{E}_{y \sim \rho(\cdot|x)} \left[g^{\theta}_{\text{F-KL}}(x, y) \right] - \mathbb{E}_{y \sim \rho'(\cdot|x)} \left[g^{\theta}_{\text{F-KL}}(x, y) \right] \right\|_2 \\ &\quad \leq 2\sqrt{2} B \, \mathrm{TV}(\rho(\cdot \mid x), \rho'(\cdot \mid x)) \,. \end{aligned} \tag{35}$$

Finally, using Equation 25 and moving the norm inside the expectation over prompts gives

$$\left\| \bar{\nabla}_\theta \mathcal{L}_{D_{\text{F-KL}}}(\theta; \rho) - \bar{\nabla}_\theta \mathcal{L}_{D_{\text{F-KL}}}(\theta; \rho') \right\|_2 \tag{36}$$

$$= \left\| \mathbb{E}_{x \sim p_{\text{data}}} \left[\mathbb{E}_{y \sim \rho(\cdot|x)} \left[g^{\theta}_{\text{F-KL}}(x, y) \right] - \mathbb{E}_{y \sim \rho'(\cdot|x)} \left[g^{\theta}_{\text{F-KL}}(x, y) \right] \right] \right\|_2 \tag{37}$$

$$\leq \mathbb{E}_{x \sim p_{\text{data}}} \left[\left\| \mathbb{E}_{y \sim \rho(\cdot|x)} \left[g^{\theta}_{\text{F-KL}}(x, y) \right] - \mathbb{E}_{y \sim \rho'(\cdot|x)} \left[g^{\theta}_{\text{F-KL}}(x, y) \right] \right\|_2 \right] \tag{38}$$

$$\leq 2\sqrt{2} B \, \mathbb{E}_{x \sim p_{\text{data}}} \left[\mathrm{TV}(\rho(\cdot \mid x), \rho'(\cdot \mid x)) \right], \tag{39}$$

which proves Equation 27. The first inequality above is the triangle inequality for an expectation; the second is the fixed-prompt bound derived from Equation 34.

For the second claim, Equation 33 holds for every x and y, so

$$\left\| g^{\theta}_{\text{F-KL}}(x, y) \right\|_2^2 \leq 2B^2. \tag{40}$$

Taking the expectation of both sides under $x \sim p_{\text{data}}$ and $y \sim \rho(\cdot \mid x)$ proves Equation 28. □

Interpretation. The first conclusion of Theorem C.1 states that changing the rollout policy can change the expected forward-KL update only in proportion to the total-variation distance between the resulting completion distributions. Its constant does not depend on student–teacher probability ratios. Thus, when two rollout policies generate similar distributions over completions, their expected forward-KL updates must also be similar. This is a worst-case stability statement rather than rollout invariance: the guarantee can be loose when the two completion distributions have total variation close to one, or when the Jacobian bound B is large.

The second conclusion bounds the second moment of the gradient contribution from an individual sampled completion. In particular, the forward-KL objective cannot produce arbitrarily large gradient contributions solely because the student assigns very little probability to a token preferred by the teacher. This distinguishes the gradient from the forward-KL objective value, which is itself unbounded. The result also bounds the variance by

$$\mathbb{E}\left[\left\| g^{\theta}_{\text{F-KL}} - \mathbb{E}[g^{\theta}_{\text{F-KL}}] \right\|_2^2 \right] \leq 2B^2. \tag{41}$$

These conclusions provide a possible explanation for the empirical robustness of forward KL in Section 5. Across the rollout-policy spectrum, forward-KL training attains similar final performance despite substantial changes in the source of the generated trajectories. The theorem shows that this behaviour is compatible with a uniformly controlled change in the expected update, and that extreme student–teacher likelihood ratios do not by themselves create unbounded forward-KL logit gradients. This is also consistent with the gradient-clipping ablation in Section 6, where forward KL remains substantially more stable than reverse KL when clipping is removed.

The theorem does not, however, prove that different rollout policies must produce similar models or similar final performance. It controls a one-step expected semi-gradient, and its bound depends on both the distance between the complete-trajectory distributions and the largest singular value B of the student-logit Jacobian. Nor does it directly explain the observed differences in catastrophic forgetting or parameter-update sparsity, which the experiments show are governed primarily by the learning rate. The empirical results should therefore be interpreted as consistent with the stability result, rather than as a direct consequence of it.

C.3 Reverse KL admits no distribution-free rollout-stability bound

At a fixed prefix h_n, define the student–teacher log-likelihood ratio for token $v \in \mathcal{V}$ by

$$r_n(v) := \log \frac{\pi_S^\theta(v \mid h_n)}{\pi_T(v \mid h_n)}. \tag{42}$$

We write $r_n = (r_n(v))_{v\in\mathcal{V}}$ for the corresponding vector over the vocabulary. The reverse-KL gradient with respect to the student logits is

$$\begin{aligned}&\nabla_{z_S^\theta(h_n)} D_{\text{R-KL}}\left(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\right) \\ &\quad = \pi_S^\theta(\cdot \mid h_n) \odot \left[r_n - D_{\text{R-KL}}\left(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\right)\mathbf{1}\right].\end{aligned} \tag{43}$$

All operations in Equation 43 are coordinate-wise. Analogously to Equation 24, define the reverse-KL parameter-gradient contribution of a complete sampled trajectory as

$$\begin{aligned}&g_{\text{R-KL}}^\theta(x, y) \\ &:= \frac{1}{L_y}\sum_{n=1}^{L_y} J_\theta(h_n)^\top \left\{\pi_S^\theta(\cdot \mid h_n) \odot \left[r_n - D_{\text{R-KL}}\left(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\right)\mathbf{1}\right]\right\}.\end{aligned} \tag{44}$$

The reverse-KL semi-gradient is therefore

$$\bar{\nabla}_\theta \mathcal{L}_{D_{\text{R-KL}}}(\theta; \rho) = \mathbb{E}_{x\sim p_{\text{data}}} \mathbb{E}_{y\sim\rho(\cdot|x)}\left[g_{\text{R-KL}}^\theta(x, y)\right]. \tag{45}$$

Unlike its forward-KL counterpart, the vector in Equation 43 is not uniformly bounded over student and teacher distributions.

Theorem C.2 (No distribution-free reverse-KL rollout-stability bound)**.** *There is no finite constant C, independent of the student and teacher token distributions, for which*

$$\begin{aligned}&\left\|\bar{\nabla}_\theta \mathcal{L}_{D_{\text{R-KL}}}(\theta; \rho) - \bar{\nabla}_\theta \mathcal{L}_{D_{\text{R-KL}}}(\theta; \rho')\right\|_2 \\ &\quad \le C\, \mathbb{E}_{x\sim p_{\text{data}}}\left[\text{TV}(\rho(\cdot \mid x), \rho'(\cdot \mid x))\right]\end{aligned} \tag{46}$$

holds for all rollout policies ρ and ρ'. This remains true for a two-token vocabulary and a direct logit parameterization whose Jacobian has largest singular value one.

Proof. It is sufficient to construct a counterexample for one fixed prompt x. Consider two length-two completions, denoted by $y^{(0)}$ and $y^{(\delta)}$. At the initial prefix, which is shared by both completions, let the student and teacher distributions coincide. Their reverse-KL gradient at that prefix is therefore zero. At the second prefix of $y^{(0)}$, again let the two distributions coincide. Consequently,

$$g_{\text{R-KL}}^\theta(x, y^{(0)}) = 0. \tag{47}$$

At the second prefix of $y^{(\delta)}$, let the vocabulary contain two tokens and set

$$\pi_S^\theta(\cdot \mid h_2) = \left(\tfrac{1}{2}, \tfrac{1}{2}\right), \qquad \pi_T(\cdot \mid h_2) = (\delta, 1-\delta), \qquad 0 < \delta < \tfrac{1}{2}. \tag{48}$$

Use a direct logit parameterization, so that $J_\theta(h_2)^\top = I$. If

$$r_1 = \log \frac{1/2}{\delta}, \qquad r_2 = \log \frac{1/2}{1-\delta}, \tag{49}$$

then

$$D_{\text{R-KL}}\left(\pi_S^\theta(\cdot \mid h_2), \pi_T(\cdot \mid h_2)\right) = \frac{r_1 + r_2}{2}. \tag{50}$$

Substituting into Equation 43 gives

$$\begin{aligned}&\nabla_{z_S^\theta(h_2)} D_{\text{R-KL}}\left(\pi_S^\theta(\cdot \mid h_2), \pi_T(\cdot \mid h_2)\right) \\ &\quad = \frac{1}{4}\log\left(\frac{1-\delta}{\delta}\right)(1, -1).\end{aligned} \tag{51}$$

The first-token contribution is zero and $L_y = 2$. Therefore,

$$\left\| g^{\theta}_{\text{R-KL}}(x, y^{(\delta)}) \right\|_2 = \frac{1}{2} \left\| \nabla_{z^{\theta}_S(h_2)} D_{\text{R-KL}} \left(\pi^{\theta}_S(\cdot \mid h_2), \pi_T(\cdot \mid h_2) \right) \right\|_2 \tag{52}$$

$$= \frac{\sqrt{2}}{8} \left| \log \left(\frac{1-\delta}{\delta} \right) \right|, \tag{53}$$

which diverges as $\delta \downarrow 0$.

Let Δ_y denote the point mass on completion y. For any fixed $\varepsilon \in (0, 1)$, define

$$\rho'(\cdot \mid x) = \Delta_{y^{(0)}}, \qquad \rho(\cdot \mid x) = (1-\varepsilon)\Delta_{y^{(0)}} + \varepsilon \Delta_{y^{(\delta)}}. \tag{54}$$

Their total-variation distance is

$$\text{TV}(\rho(\cdot \mid x), \rho'(\cdot \mid x)) = \varepsilon. \tag{55}$$

Using Equations 45, 47, and 54, the difference between their expected reverse-KL updates is

$$\left\| \bar{\nabla}_{\theta} \mathcal{L}_{D_{\text{R-KL}}}(\theta; \rho) - \bar{\nabla}_{\theta} \mathcal{L}_{D_{\text{R-KL}}}(\theta; \rho') \right\|_2$$

$$= \varepsilon \left\| g^{\theta}_{\text{R-KL}}(x, y^{(\delta)}) \right\|_2 \tag{56}$$

$$= \frac{\varepsilon\sqrt{2}}{8} \left| \log \left(\frac{1-\delta}{\delta} \right) \right|. \tag{57}$$

The rollout total-variation distance remains equal to ε, while Equation 57 becomes arbitrarily large as $\delta \downarrow 0$. Hence no universal constant C can satisfy Equation 46. □

The construction requires no zero-probability token: every teacher probability is strictly positive for $\delta > 0$. It therefore applies to softmax language models. The theorem states that proximity between two rollout distributions alone cannot uniformly control the difference between their expected reverse-KL updates. It does not assert that reverse-KL training is always unstable or that either on- or off-policy rollouts are universally preferable.

C.4 A REVERSE-KL BOUND UNDER CONTROLLED LIKELIHOOD RATIOS

The absence of a distribution-free guarantee does not preclude a conditional stability result. Such a result follows when the range of the token-level student–teacher log-likelihood ratios is bounded.

Corollary C.3 (Reverse-KL stability under a bounded log-ratio range)**.** *Suppose that the Jacobian condition in Equation 26 holds. In addition, suppose that there exists $R < \infty$ such that, at every prefix occurring under ρ or ρ',*

$$\max_{v \in \mathcal{V}} \log \frac{\pi^{\theta}_S(v \mid h_n)}{\pi_T(v \mid h_n)} - \min_{v \in \mathcal{V}} \log \frac{\pi^{\theta}_S(v \mid h_n)}{\pi_T(v \mid h_n)} \le R. \tag{58}$$

Then

$$\left\| \bar{\nabla}_{\theta} \mathcal{L}_{D_{\text{R-KL}}}(\theta; \rho) - \bar{\nabla}_{\theta} \mathcal{L}_{D_{\text{R-KL}}}(\theta; \rho') \right\|_2$$

$$\le 2BR\, \mathbb{E}_{x \sim p_{\text{data}}} \left[\text{TV}(\rho(\cdot \mid x), \rho'(\cdot \mid x)) \right]. \tag{59}$$

Moreover, for every rollout policy ρ,

$$\mathbb{E}_{x \sim p_{\text{data}}} \mathbb{E}_{y \sim \rho(\cdot|x)} \left[\left\| g^{\theta}_{\text{R-KL}}(x, y) \right\|_2^2 \right] \le B^2 R^2. \tag{60}$$

Proof. For a fixed prefix, let

$$\overline{r}_n := \sum_{v \in \mathcal{V}} \pi^{\theta}_S(v \mid h_n) r_n(v) = D_{\text{R-KL}} \left(\pi^{\theta}_S(\cdot \mid h_n), \pi_T(\cdot \mid h_n) \right). \tag{61}$$

Because $\overline{r}_n$ is a weighted average of the coordinates of r_n, it lies between their minimum and maximum. The assumption in Equation 58 therefore gives

$$|r_n(v) - \overline{r}_n| \le R \qquad \text{for every } v \in \mathcal{V}. \tag{62}$$

Using Equation 43,

$$\begin{aligned}
&\left\|\nabla_{z_S^\theta(h_n)} D_{\text{R-KL}}\left(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\right)\right\|_2 \\
&\quad \le \left\|\pi_S^\theta(\cdot \mid h_n) \odot (r_n - \overline{r}_n \mathbf{1})\right\|_1 && (63)\\
&\quad = \sum_{v \in \mathcal{V}} \pi_S^\theta(v \mid h_n)\, |r_n(v) - \overline{r}_n| && (64)\\
&\quad \le R. && (65)
\end{aligned}$$

The Jacobian condition now implies that the norm of each token-level parameter gradient is at most BR. Applying the triangle inequality to the token average in Equation 44 gives

$$\left\|g_{\text{R-KL}}^\theta(x, y)\right\|_2 \le BR. \tag{66}$$

Applying the total-variation argument from the proof of Theorem C.1 with $g(y) = g_{\text{R-KL}}^\theta(x, y)$ proves Equation 59. Squaring Equation 66 and taking expectations proves Equation 60. □

Interpretation. Theorem C.2 provides a worst-case distinction between the two KL directions. For forward KL, rollout proximity alone controls the difference between expected updates, subject only to the common Jacobian bound. For reverse KL, the same rollout proximity is insufficient: a rollout can place even a small amount of probability on completions containing a prefix at which the student assigns substantial mass to a teacher-improbable token, producing an arbitrarily large change in the expected update. The two factors therefore play distinct and complementary roles: the rollout policy determines which prefixes are visited and how strongly they are weighted, while the local student–teacher log-likelihood ratios determine how large their reverse-KL gradient contributions can be. A rollout-probability change of order ε can consequently be amplified by a local gradient of order $\log(1/\delta)$, producing an update difference of order $\varepsilon \log(1/\delta)$.

Extreme likelihood ratios may already be present at initialization, but the result should not be interpreted solely as a statement about poor initialization or unusually large student logits. In the counterexample, the student distribution is uniform; the gradient becomes large because the teacher assigns vanishing probability to a student-supported token. Such disagreement may instead emerge during training or occur only at particular prefixes that are emphasized by one rollout policy.

This distinction is consistent with the results in Section 5, where reverse-KL performance varies substantially across the rollout-policy spectrum and is less stable than forward-KL performance. It is also consistent with the gradient-clipping ablation in Section 6: removing clipping disproportionately harms reverse KL, whose logit gradient contains an unbounded student–teacher log-likelihood ratio. Global gradient clipping bounds the realized gradient norm and can therefore mitigate this failure mode, although it does not impose the likelihood-ratio condition in Corollary C.3 or guarantee similar update directions under different rollout policies.

Neither result proves that reverse KL must be unstable in a particular run, or that student-generated rollouts must outperform teacher-generated rollouts. The empirical findings should instead be interpreted as instances of the greater sensitivity permitted by the reverse-KL gradient. The bounded-ratio corollary additionally predicts that reverse-KL sensitivity should be associated with prefixes having extreme student–teacher log-likelihood ratios, suggesting a directly testable diagnostic for the observed training instabilities.

C.5 Relation to sequence-level KL divergences

There is also an exact structural relationship between KL direction and rollout source. To state it cleanly, consider a fixed prompt x and completions with a fixed generation horizon H, where H denotes the number of generated token positions. Variable-length completions can be represented by including an end-of-sequence token and treating the terminated state as absorbing up to position H.

Using the prefix notation $h_n = (x, y_{<n})$, the teacher and student induce the sequence distributions

$$\pi_T(y \mid x) := \prod_{n=1}^{H} \pi_T(y_n \mid h_n), \qquad \pi_S^\theta(y \mid x) := \prod_{n=1}^{H} \pi_S^\theta(y_n \mid h_n). \tag{67}$$

Their marginal distributions over the prefix preceding token n are

$$\pi_T(y_{<n} \mid x) := \prod_{m=1}^{n-1} \pi_T(y_m \mid h_m), \qquad \pi_S^\theta(y_{<n} \mid x) := \prod_{m=1}^{n-1} \pi_S^\theta(y_m \mid h_m). \tag{68}$$

For $n = 1$, these are empty products, so both distributions place unit mass on the empty output prefix.

Proposition C.4 (Matched rollouts recover sequence-level KL)**.** *For every prompt x,*

$$D_{\mathrm{KL}}\big(\pi_T(\cdot \mid x) \| \pi_S^\theta(\cdot \mid x)\big) = \sum_{n=1}^{H} \mathbb{E}_{y_{<n} \sim \pi_T(\cdot|x)} \left[D_{\text{F-KL}}\big(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\big)\right], \tag{69}$$

$$D_{\mathrm{KL}}\big(\pi_S^\theta(\cdot \mid x) \| \pi_T(\cdot \mid x)\big) = \sum_{n=1}^{H} \mathbb{E}_{y_{<n} \sim \pi_S^\theta(\cdot|x)} \left[D_{\text{R-KL}}\big(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\big)\right]. \tag{70}$$

Proof. For the forward direction, the autoregressive factorization gives

$$\begin{aligned}
&D_{\mathrm{KL}}\big(\pi_T(\cdot \mid x) \| \pi_S^\theta(\cdot \mid x)\big) \\
&= \mathbb{E}_{y \sim \pi_T(\cdot|x)} \left[\log \frac{\pi_T(y \mid x)}{\pi_S^\theta(y \mid x)} \right] && (71)\\
&= \mathbb{E}_{y \sim \pi_T(\cdot|x)} \left[\sum_{n=1}^{H} \log \frac{\pi_T(y_n \mid h_n)}{\pi_S^\theta(y_n \mid h_n)} \right] && (72)\\
&= \sum_{n=1}^{H} \mathbb{E}_{y_{<n} \sim \pi_T(\cdot|x)} \left[\mathbb{E}_{y_n \sim \pi_T(\cdot|h_n)} \left[\log \frac{\pi_T(y_n \mid h_n)}{\pi_S^\theta(y_n \mid h_n)} \right] \right] && (73)\\
&= \sum_{n=1}^{H} \mathbb{E}_{y_{<n} \sim \pi_T(\cdot|x)} \left[D_{\text{F-KL}}\left(\pi_S^\theta(\cdot \mid h_n), \pi_T(\cdot \mid h_n)\right)\right]. && (74)
\end{aligned}$$

The third equality first conditions on the prefix $y_{<n}$ and then averages over the teacher distribution of the next token. This proves Equation 69. Exchanging the student and teacher distributions proves Equation 70. □

Proposition C.4 identifies teacher rollouts with forward KL and student rollouts with reverse KL as the two chain-rule decompositions of sequence-level KL. The crossed pairings do not, in general, correspond to either sequence-level KL divergence. This is an identity between objective values, not an ordering of the objectives or their optimization properties. In particular, under student rollouts, we stop gradients through the sampled prefixes. Consequently, although the expected OnPD reverse-KL objective equals sequence-level reverse KL in the fixed-horizon, summed-loss setting, the update used in training is only a semi-gradient and need not equal the total gradient of that sequence-level divergence.

Finally, Proposition C.4 uses sums of token-level divergences. For fixed H, the corresponding token-averaged objectives are exactly $1/H$ times the sequence-level KL divergences. In the experiments, however, each variable-length completion is normalized by its realized length L_y. The implemented objective is therefore a length-weighted token-level divergence and is not generally a constant multiple of sequence-level KL.

D EXPERIMENTAL DETAILS

D.1 DATASETS

D.1.1 TRAINING DATASETS.

We train on three reasoning datasets, each divided into a training split and a disjoint held-out evaluation split. Every training example consists of a task prompt and a worked demonstration. We use the task-specific prompts in Table 1. During teacher supervised fine-tuning (SFT), the vanilla task prompt is provided as input and the demonstration is used as the target completion. The held-out demonstrations are never used for training or provided during evaluation.

COUNTDOWN-3. We procedurally generated 5,000 training and 500 evaluation candidates using random seeds 42 and 43, respectively. Each problem contains three operands sampled with replacement from $\{1, \ldots, 10, 12, 15, 20, 25, 50, 75\}$ and a target obtained by combining all operands through addition, subtraction, multiplication, or exact integer division. We queried GPT-5 for worked demonstrations and retained only correctly solved, properly tagged examples. This produced 4,163 training examples and 426 held-out evaluation examples.

SCIENCE. The Science dataset contains four-option multiple-choice questions from SciKnowEval. Its fixed splits contain 2,674 training examples and 507 held-out evaluation examples. Each training example includes a worked reasoning trace followed by the correct option letter in an `<answer>` block.

MEDREASON. MedReason consists of medical multiple-choice questions from `UCSC-VLAA/MedReason`. From the prepared source partition, we selected candidate pools of 5,000 training and 500 evaluation questions. We generated worked demonstrations with GPT-5 and retained only responses that were properly formatted and selected the ground-truth option. The resulting splits contain 4,748 training examples and 471 held-out evaluation examples.

Dataset	System prompt	User prompt
COUNTDOWN-3	You are a careful arithmetic solver.	Solve the Countdown arithmetic problem. Using the provided numbers, create an equation that equals the target. You may use the operations $+$, $-$, $\times$, and $/$, and each number must be used exactly once; every division must have an integer result. Show your work in `<think>...</think>` tags and return the solution as compact, comma-separated equations in `<answer>...</answer>` tags. For example, `<answer>2+3=5,5*4=20</answer>`. The question is appended as `Numbers: [...]` and `Target: ...`.
SCIENCE	You are a careful solver of multiple-choice science questions.	Given a question and four options, select the correct answer. Reason step by step in `<think>...</think>` tags and place only the corresponding option letter (A, B, C, or D) in `<answer>...</answer>` tags. The question and answer choices are appended to this instruction.
MEDREASON	You are a careful medical reasoning assistant.	Given a medical multiple-choice question and its answer choices, select the correct answer. Reason step by step in `<think>...</think>` tags and place only the letter corresponding to the correct option in `<answer>...</answer>` tags; do not repeat the answer text. The question and answer choices are appended to this instruction.

Table 1: Task prompts used for teacher SFT and policy-distillation experiments. During SFT, the worked demonstration is supplied as the target completion rather than included in the input prompt.

In-Distribution Evaluation Procedure. To ensure computational efficiency, unless otherwise stated for our evaluations we use the first 200 examples from each dataset, giving 600 generated responses per evaluated checkpoint. We generate one response per prompt from the student at temperature 0.5 and top-$p = 1.0$. The maximum prompt length is 2,048 tokens for Countdown-3 and

Science and 4,096 tokens for MedReason; the maximum completion length is 2,048 tokens for all three datasets. We report the accuracy on the dataset matching the student's training dataset. Unless otherwise stated, the in-distribution evaluation uses temperature 0.5.

D.1.2 OOD Evaluation Datasets.

To measure out-of-distribution (OOD) capability retention, we evaluate our trained student models on 7 benchmarks spanning three broad categories: knowledge and factual reliability (MMLU-Pro (Wang et al., 2024b) and TruthfulQA (Lin et al., 2021)), instruction following and code generation (IFEval (Zhou et al., 2023) and HumanEval (Chen et al., 2021)), and social reasoning and safety (EQ-Bench (Paech, 2024), BBQ (Parrish et al., 2021), and ToxiGen (Hartvigsen et al., 2022)).

We use the interface provided by the Evaluation Harness (Gao et al., 2024), ensuring a standardised evaluation protocol. We evaluate 200 examples each from TruthfulQA-MC1, IFEval, ToxiGen, and BBQ; all 164 HumanEval-Instruct examples; and all 171 EQ-Bench examples. For MMLU-Pro, the 200-example limit is applied independently to each of its 14 subject areas, yielding 2,800 examples. In total, this gives 3,935 examples across seven benchmarks. We use each benchmark's standard metric: extracted exact match for MMLU-Pro, accuracy for TruthfulQA-MC1, ToxiGen, and BBQ, pass@1 for HumanEval-Instruct, strict prompt-level accuracy for IFEval, and the normalized EQ-Bench score. The aggregate OOD score is the unweighted mean of these seven task scores.

D.2 Models

D.2.1 Student models

All primary experiments use `Meta-Llama-3.2-1B-Instruct` as the student model. Each run begins from the same checkpoint and trains a separate student for each dataset, data-collection policy, KL direction, learning rate, and random seed. The student is fully fine-tuned rather than adapted with LoRA; all teacher parameters remain frozen. Each student is paired with the teacher from the same family described above. These models use a compatible tokenizer and vocabulary, allowing the forward- and reverse-KL objectives to compare their token distributions directly. Both models receive the same vanilla task prompt and chat-template formatting. The distinction between on- and off-policy training is therefore solely the model used to generate the training trajectory: the student generates trajectories for on-policy distillation, whereas the frozen teacher generates them for off-policy distillation.

D.2.2 Teacher Model Training

We construct a separate teacher for each dataset and model family using two stages: supervised fine-tuning (SFT) on worked demonstrations, followed, where applicable, by reinforcement learning with Group Relative Policy Optimization (GRPO). Teachers are therefore task-specific rather than jointly trained across datasets.

Supervised fine-tuning. We first fine-tune each teacher backbone on the training split of the corresponding dataset. The input is the vanilla task prompt from Table 1, without a demonstration in context, and the target completion is the associated worked demonstration. Thus, SFT trains the teacher to reproduce both the reasoning trace and the formatted final answer. For Countdown-3 and MedReason, these targets are the verified GPT-5 demonstrations described above; for Science, we use the worked demonstrations included in the processed training split.

We train using token-level cross-entropy over the demonstration tokens for 500 optimizer steps. We use a learning rate of 5×10^{-5}, a cosine learning-rate schedule with 10 warm-up steps, an effective batch size of 32, and gradient clipping at norm 1.0. Prompts and completions are truncated to at most 2,048 tokens each. For computational efficiency, all teachers are adapted with LoRA of rank 256 and scaling parameter 256, applied to the query, key, value, output, gate, up, and down projection matrices, with no LoRA dropout or bias. Separate SFT runs are performed for Countdown-3, Science, and MedReason.

GRPO refinement. For Science and MedReason, where the performance after the SFT stage has not yet saturated as in the case of Countdown-3, we further refine the dataset-specific SFT checkpoints

Table 2: In-distribution performance of the trained teachers. Results use 200 test examples per dataset, temperature 0.5, one sampled response per prompt, and a 2,048-token completion limit. Response lengths are reported as mean ± standard deviation.

Teacher	Dataset	Accuracy (%)	Response length (tokens)
Qwen2.5-7B	MedReason	77.5	90.95 ± 28.58
	Science	72.0	288.82 ± 90.38
	Countdown-3	95.5	168.42 ± 316.40
Llama-3.1-8B	MedReason	85.5	93.74 ± 30.81
	Science	67.5	134.65 ± 50.77
	Countdown-3	99.0	98.82 ± 78.54

using GRPO. GRPO receives only the vanilla task prompt and samples eight candidate completions per prompt at temperature 1.0 and nucleus-sampling threshold $p = 0.95$. The reward is the sum of task correctness and formatting rewards. Correct answers receive a reward of 2.0, using exact option accuracy for Science and MedReason. Additional rewards encourage exactly one pair of `<think>` and `<answer>` tags, the correct ordering of these tags, and strict adherence to the required response structure; each formatting component contributes at most 0.5. We use group-normalized rewards with the DAPO loss and no reference-model KL penalty ($\beta = 0$).

GRPO training runs for 500 optimizer steps with a learning rate of 10^{-5}, a cosine schedule, a warm-up ratio of 0.1, an eight-bit AdamW optimizer, and gradient clipping at norm 1.0. The per-device batch size is 8 with 16 gradient-accumulation steps. Each rollout batch contains 32 completions for Science and 48 for MedReason. The maximum prompt and completion lengths are 2,048 and 824 tokens, respectively. GRPO also uses rank-256 LoRA with scaling parameter 256 over the same projection matrices as SFT; models are loaded in four-bit precision, while optimization uses bfloat16. Generation is performed with the colocated vLLM backend.

Selected teachers. Teachers are paired with students from the same model family; Llama-3.1 teachers are paired with Llama-3.2 students because they share a tokenizer and vocabulary. For Science and MedReason, the selected teachers are the GRPO-refined checkpoints, generally at step 500; the selected Qwen2.5 MedReason and Llama-3 Science checkpoints are taken at step 450 due to better performance. For Countdown-3, we use the step-500 SFT checkpoint directly, without an additional GRPO stage. We define the fixed teacher policy as $\pi_T(\cdot \mid h) = \mathrm{softmax}(z_T(h)/0.5)$ and the student policy as $\pi_S^\theta(\cdot \mid h) = \mathrm{softmax}(z_S^\theta(h)/1.0)$. These definitions are used consistently both for trajectory generation and when computing the token-level KL objective. Accordingly, OffPD samples trajectories from π_T, whereas OnPD samples trajectories from π_S^θ; in both conditions, the loss compares the same π_T and π_S^θ at the resulting prefixes. Temperature is therefore not varied as an additional experimental factor between OnPD and OffPD: the sole intervention is whether the rollout distribution is π_T or π_S^θ. We use temperature 0.5 for the teacher because preliminary sampling at temperature 1.0 occasionally produced degenerate continuations near the end of long trajectories. We ablate the effect of this choice in Appendix A.12. We characterise the performance of the selected teacher models after this training procedure in Table 2.

D.3 Training Details

The most important hyperparameters are presented in Table 3. We fully fine-tune the student for 150 optimizer steps, using one sampled completion per prompt. We choose the 150-step training budget based on the learning curves in Figure 19, which show that target-task performance and forgetting have largely stabilised by this point. We use the same training budget across configurations. Each optimizer step uses an effective batch of 32 prompts, implemented with a per-device batch size of 1 and 32 gradient-accumulation steps. Prompts are sampled without replacement within each batch, but may recur across optimizer steps. We use fused AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, no weight decay, and gradient clipping at norm 1.0. The learning rate is held constant following 10 linear warm-up steps. Our main experiments consider learning rates in $\{1 \times 10^{-5}, 5 \times 10^{-5}\}$; the dedicated Countdown-3 learning-rate study additionally considers $\{2, 3, 4, 6\} \times 10^{-5}$. Results are reported over random seeds 42, 43, and 44.

Hyperparameter	Value	Notes
Optimization steps	150	Reported primary comparisons use the checkpoint at step 150.
Optimizer	Fused AdamW	$\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$.
Weight decay	0	
Learning-rate schedule	Constant with linear warm-up	10 warm-up steps.
Per-device batch size	1	
Gradient accumulation	32 steps	Gives an effective batch size of 32 prompts per optimizer step.
Trajectories per prompt	1	One completion is sampled for every selected prompt.
Maximum gradient norm	1.0	
Student temperature	1.0	Used for student sampling and student logits in the KL objective.
Teacher temperature	0.5	Used for teacher sampling and teacher logits in the KL objective.
On-policy trajectory source	Student	Sampled at temperature 1.0.
Off-policy trajectory source	Teacher	Sampled at temperature 0.5.
Top-p	1.0	Used for both on- and off-policy trajectory sampling.
Maximum prompt length	2,048 tokens	Shared across datasets.
Maximum completion length	512 / 1,024 tokens	512 for Countdown-3; 1,024 for Science and MedReason.
Loss reduction	Mean over completion tokens	Computed over the full vocabulary at every non-padding completion position.
Gradient checkpointing	On-policy: enabled; off-policy: disabled	This difference is used for memory management.

Table 3: Principal hyperparameters used for distillation training.

We train with either forward KL, $\mathrm{KL}(p_{\text{teacher}} \| p_{\text{student}})$, or reverse KL, $\mathrm{KL}(p_{\text{student}} \| p_{\text{teacher}})$. The student and teacher distributions use temperatures 1.0 and 0.5, respectively. The loss is computed over the full vocabulary at every non-padding completion position and then averaged over completion tokens and examples. The teacher remains frozen throughout training.

For on-policy distillation, trajectories are sampled from the student at temperature 1.0. For off-policy distillation, trajectories are sampled from the teacher at temperature 0.5. Both use $p = 1.0$ nucleus sampling and vanilla task prompts without demonstrations. The maximum prompt length is 2,048 tokens. Maximum completion lengths are 512 tokens for Countdown-3 and 1,024 tokens for Science and MedReason. Gradient checkpointing is enabled for on-policy runs and disabled for off-policy runs. Students are trained without LoRA, so all student parameters are updated.

D.4 Parameter-Update Sparsity

Following Mukherjee et al. (2025), we measure update sparsity as the fraction of model parameters whose absolute change during training falls below a fixed threshold. Let θ^{init} and θ^{final} denote the initial and final student parameters, and let P be the total number of parameters. We compute

$$S_\tau = \frac{1}{P} \sum_{i=1}^{P} \mathbb{I}\left[\left|\theta_i^{\text{final}} - \theta_i^{\text{init}}\right| < \tau\right].$$

All students are fully fine-tuned, so this comparison includes the entire model rather than only an adapter or a subset of trainable parameters. Higher values indicate that a larger fraction of parameters remain close to their initial values; they do not necessarily indicate exactly zero updates.

We use $\tau = 10^{-6}$ for the main results. Appendix A.6 reports the corresponding analysis at $\tau = 10^{-8}$, which preserves the main qualitative conclusions.

D.5 Implementation of the Rollout-Policy Spectrum

At each generation step, the student and teacher are evaluated on the same prefix. The unconstrained rollout-policy score is

$$(\hat{z}_\lambda)_v = \frac{1}{2}\left(\log \pi_S(v) + \log \pi_T(v)\right) + \frac{\lambda}{2}\left(\log \pi_S(v) - \log \pi_T(v)\right) \tag{75}$$

$$= \frac{1+\lambda}{2}\log \pi_S(v) + \frac{1-\lambda}{2}\log \pi_T(v). \tag{76}$$

The corresponding policy is obtained by normalising these scores:

$$\pi_\lambda(v) = \frac{\exp((\hat{z}_\lambda)_v)}{\sum_{u\in\mathcal{V}} \exp((\hat{z}_\lambda)_u)}.$$

Thus, $\lambda = -1$ recovers π_T, $\lambda = 1$ recovers π_S, and $\lambda = 0$ gives their normalised geometric midpoint.

For $|\lambda| > 1$, one of the coefficients above becomes negative. This can disproportionately amplify tokens that are unlikely under both policies but relatively less unlikely under the favoured policy. We prevent this using one-sided log-ratio clipping and a plausibility mask, similar in spirit to Li et al. (2023).

Define the token-level log-likelihood ratio

$$r_v = \log \pi_S(v) - \log \pi_T(v),$$

and the student and teacher plausibility masks

$$M_S(v) = \mathbb{I}\left[\log \pi_S(v) \geq \max_{u\in\mathcal{V}} \log \pi_S(u) + \log \alpha\right], \tag{77}$$

$$M_T(v) = \mathbb{I}\left[\log \pi_T(v) \geq \max_{u\in\mathcal{V}} \log \pi_T(u) + \log \alpha\right]. \tag{78}$$

A token is therefore eligible for contrastive amplification only if its probability under the favoured policy is at least an α fraction of that policy's maximum token probability.

We define the clipped student- and teacher-favoured bonuses as

$$b_S(v) = M_S(v)\min\{\max\{r_v, 0\}, c\}, \tag{79}$$

$$b_T(v) = M_T(v)\min\{\max\{-r_v, 0\}, c\}, \tag{80}$$

where $c > 0$ is the clipping threshold. The safeguarded sampling scores are then

$$(\tilde{z}_\lambda)_v = \begin{cases} (\hat{z}_\lambda)_v, & |\lambda| \leq 1, \\ \log \pi_S(v) + \dfrac{\lambda - 1}{2} b_S(v), & \lambda > 1, \\ \log \pi_T(v) + \dfrac{-\lambda - 1}{2} b_T(v), & \lambda < -1. \end{cases}$$

The final rollout policy is

$$\pi_\lambda(v) = \mathrm{softmax}(\tilde{z}_\lambda)_v.$$

The plausibility masks gate only the additional extrapolation bonus: tokens outside the mask retain their probability under the corresponding anchor policy but receive no amplification. This preserves $\pi_{-1} = \pi_T$, $\pi_1 = \pi_S$, and continuity at the boundaries of the interpolation region. In our experiments, we use

$$\alpha = 0.2, \qquad c = 2\log 3 \approx 2.2.$$

For the range $\lambda \in [-2, 2]$, this limits the maximum pairwise-odds amplification relative to the anchor policy to a factor of three.

D.6 RLVR AFTER DISTILLATION

Purpose and initialisation. We test whether the differences induced by distillation affect subsequent RLVR on a harder task. We initialise Llama-3.2-1B-Instruct from the step-150 Countdown-3 checkpoints for all combinations of OnPD/OffPD, forward/reverse KL, and distillation learning rates $\{1\times10^{-5}, 5\times10^{-5}\}$. Each configuration has three checkpoints from distillation seeds 42, 43, and 44, giving 24 RLVR runs. We train each checkpoint on Countdown-4 with the same RLVR configuration. The RLVR seed is fixed at 42, so the three trajectories per configuration reflect different distillation seeds rather than independently varied RLVR seeds.

Training data and reward. We use 5,000 procedurally generated Countdown-4 problems, with four operands sampled with replacement from $\{1, \ldots, 10, 12, 15, 20, 25, 50, 75\}$. Targets lie between 1 and 10,000 and are constructed by combining all four operands through addition, subtraction, multiplication, or exact integer division. Dataset generation uses seed 42. Prompts follow the Countdown format in Table 1, rendered with the student's chat template, without demonstrations or teacher supervision. The reward is correctness-only: a completion receives 2 if the Countdown verifier reduces the operand pool to the target through valid arithmetic steps, and 0 otherwise. We add no separate formatting or length rewards.

Optimisation and generation. We use GRPO (Shao et al., 2024) with the DAPO loss (Yu et al., 2025), group-normalised rewards, and no reference-model KL penalty. Unlike the full-parameter distillation stage, RLVR trains fresh LoRA adapters on the frozen, four-bit-loaded checkpoint. We use rank and scaling parameter 256, zero dropout, and no bias adaptation, targeting the query, key, value, output, gate, up, and down projections. Training uses Unsloth with TRL's GRPO trainer and colocated vLLM generation; gradient checkpointing is enabled. Each prompt receives eight sampled completions at temperature 1.0 and top-$p = 0.95$. All runs use 300 optimisation steps and an RLVR learning rate of 2×10^{-6}, independently of the learning rate used during distillation. Full settings are summarised in Table 4.

Reported metrics. The curves in Figures 6 and 26 show the logged mean correctness reward on training rollouts after distillation at learning rates 1×10^{-5} and 5×10^{-5}, respectively, with each distilled checkpoint plotted separately and no additional smoothing. These are training rewards, not held-out accuracies: because the reward takes values 0 or 2, the corresponding fraction of correct rollouts is half the mean reward. These learning rates refer to distillation, not the fixed RLVR learning rate of 2×10^{-6}.

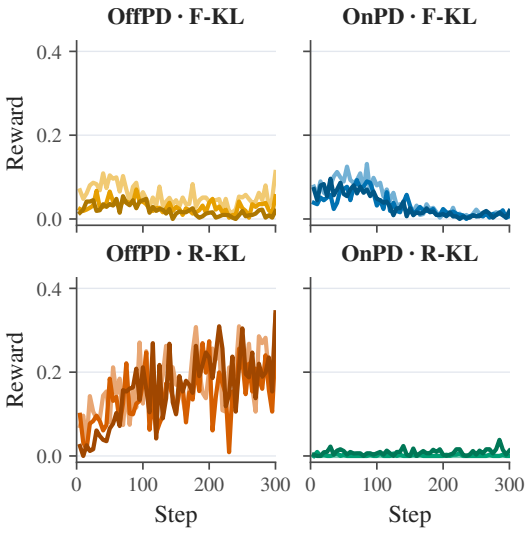


Figure 26: Countdown-4 RLVR rewards after distillation at LR 5×10^{-5}; each curve is a distillation seed.

Setting	Value	Notes
Optimisation steps	300	Starting from each step-150 distilled checkpoint.
RLVR learning rate	2×10^{-6}	Shared across all 24 runs.
Learning-rate schedule	Cosine	Warm-up over the first 10% of steps (30 steps).
Optimizer	Eight-bit AdamW	Bfloat16 training.
Trainable parameters	LoRA adapters	Rank 256, scaling parameter 256, dropout 0; no bias adaptation.
Backbone weights	Frozen, loaded in four-bit precision	Gradient checkpointing enabled.
Per-device batch size	8	Gradient accumulation over 16 steps.
Completions per prompt	8	One reward-normalisation group per prompt.
Generation batch size	48 completions	Six groups of eight completions.
Sampling	Temperature 1.0, top-p = 0.95	Colocated vLLM generation.
Maximum prompt length	2,048 tokens	Vanilla Countdown prompt with chat template.
Maximum completion length	1,024 tokens	Shared across all configurations.
Loss and reward scaling	DAPO; group normalisation	No reference-model KL penalty ($\beta = 0$).
Reward	2 for correctness, 0 otherwise	No auxiliary reward terms.
Maximum gradient norm	1.0	Global gradient clipping.
RLVR random seed	42	Distillation seeds are 42, 43, and 44.
Logging / checkpoint interval	5 / 50 steps	Reward curves show logged training rewards.

Table 4: Training settings for RLVR on Countdown-4 after Countdown-3 distillation. These settings are identical across rollout policies, KL objectives, and learning rates used in the preceding distillation stage.

Setting	Value	Notes
Student	`Qwen2.5-1.5B-Instruct`	Fully fine-tuned on Science.
Teacher	`Qwen2.5-3B-Instruct`	Frozen base instruct model; no task-specific fine-tuning.
Experimental grid	OnPD/OffPD × forward/reverse KL	Learning rates $\{1 \times 10^{-5}, 5 \times 10^{-5}\}$.
Random seeds	42	One run per condition.
Teacher conditioning	Correct demonstration and Spanish instruction	Used for teacher rollouts and teacher distributions.
Student conditioning	Vanilla Science prompt	Used for student rollouts, likelihoods, and evaluation.
Sampling temperature	1.0 for OnPD and OffPD	The conditioned teacher therefore differs from the standard OffPD temperature of 0.5.
Distribution temperature	1.0 for student and teacher	The teacher differs from the standard temperature of 0.5.
Maximum training completion length	2,048 tokens	The standard Science experiments use 1,024 tokens.
Gradient checkpointing	Enabled for OnPD and OffPD	The standard OffPD runs disable gradient checkpointing.
Style evaluation	1,000 responses per condition	200 each from Science train and test, MedReason test, GSM8K test, and Countdown-3 test.

Table 5: Experiment-specific settings for the Spanish-style transfer experiment. All unlisted training and evaluation settings follow the standard protocol.

D.7 Spanish-Style Transfer Experiment

Purpose and model setup. This experiment tests whether rollout source changes the transfer of an incidental teacher behaviour unrelated to task correctness. We distil a frozen Qwen2.5-3B-Instruct teacher into a Qwen2.5-1.5B-Instruct student on ScienceQA, conditioning the teacher to reason in Spanish. Unlike the main experiments, the teacher is not task-fine-tuned; it instead receives a correct demonstration for each training example. Deviations from the standard experimental setup are summarised in Table 5.

Teacher conditioning. For every training example, the teacher receives the Science task prompt followed by the example's correct worked demonstration and the instruction "Now answer with a response of your own, including the thinking process." We additionally append the following style instruction to the teacher's final user message:

> Only speak in Spanish. This does not apply to the special `<think></think>` and `<answer></answer>` tokens or your final answer, but any reasoning or thoughts you produce should be strictly written in Spanish.

The student never receives the demonstration or Spanish instruction. Teacher distributions are conditioned on both for every loss evaluation; OnPD samples completions from the student under the vanilla prompt, whereas OffPD samples them from the conditioned teacher.

Language annotation. We use GPT-5 to independently classify each generated response as primarily Spanish or not, treating mathematical notation, formatting tags, code, and short answer labels as language-neutral. All 1,000 responses per condition received a valid annotation. We use one judge and one training run per condition.

E Checkpoint Sparsity Analysis

In Table 6 we summarise the learning rate reported by each of the checkpoints analysed by Mukherjee et al. (2025).

Table 6: Learning rates used to produce the checkpoints in the main RL comparison and the explicit SFT comparison of Mukherjee et al. (2025). "From" denotes the checkpoint against which parameter-update sparsity is measured. Rates refer to the language model or policy actor, using the peak rate where a schedule is reported.

Stage	From	Studied checkpoint	Training method	Model/actor LR
SFT checkpoints (Appendix C)				
SFT	`meta-llama/Llama-3.1-8B`	`allenai/Llama-3.1-Tulu-3-8B-SFT`	Full-parameter SFT (Lambert et al., 2024)	5×10^{-6}
SFT	`meta-llama/Llama-3.1-70B`	`allenai/Llama-3.1-Tulu-3-70B-SFT`	Full-parameter SFT (Lambert et al., 2024)	2×10^{-6}
SFT	`Qwen/Qwen2.5-Math-7B`	`PRIME-RL/Eurus-2-7B-SFT`	Math-reasoning SFT (Cui et al., 2025)	2×10^{-5}
RL and preference-optimization checkpoints (Table 1)				
DPO	`allenai/Llama-3.1-Tulu-3-8B-SFT`	`allenai/Llama-3.1-Tulu-3-8B-DPO`	Offline DPO (Lambert et al., 2024)	5×10^{-7}
DPO	`allenai/Llama-3.1-Tulu-3-70B-SFT`	`allenai/Llama-3.1-Tulu-3-70B-DPO`	Offline DPO (Lambert et al., 2024)	5×10^{-7}
GRPO	`deepseek-ai/deepseek-math-7b-instruct`	`deepseek-ai/deepseek-math-7b-rl`	On-policy GRPO (Shao et al., 2024)	1×10^{-6}
RL-Zero	`deepseek-ai/DeepSeek-V3-Base`	`deepseek-ai/DeepSeek-R1-Zero`	On-policy GRPO (DeepSeek-AI, 2025)	Not disclosed
ORPO	`mistralai/Mistral-7B-v0.1`	`kaist-ai/mistral-orpo-beta`	Offline, reference-free ORPO (Hong et al., 2024)	8×10^{-6}
KTO	`openbmb/Eurus-7b-sft`	`openbmb/Eurus-7b-kto`	Offline KTO (Yuan et al., 2024)	5×10^{-7}
KTO	`princeton-nlp/Llama-3-Base-8B-SFT`	`princeton-nlp/Llama-3-Base-8B-SFT-KTO`	Offline KTO (Meng et al., 2024)	5×10^{-7}
PPO	`peiyi9979/mistral-7b-sft`	`peiyi9979/math-shepherd-mistral-7b-rl`	On-policy PPO (Wang et al., 2024a)	1×10^{-6}
SimPO	`meta-llama/Meta-Llama-3-8B-Instruct`	`princeton-nlp/Llama-3-Instruct-8B-SimPO`	Offline, reference-free SimPO (Meng et al., 2024)	1×10^{-6}
PRIME	`PRIME-RL/Eurus-2-7B-SFT`	`PRIME-RL/Eurus-2-7B-PRIME`	On-policy PRIME (Cui et al., 2025)	5×10^{-7}

Notes. The SFT mean and median are 9×10^{-6} and 5×10^{-6}, respectively. The RL/preference mean and median are 1.5×10^{-6} and 5×10^{-7}, calculated over the nine checkpoints with disclosed actor rates and excluding `DeepSeek-R1-Zero`. For PRIME, the separately trained implicit reward model used a learning rate of 1×10^{-6}; this value is excluded because its parameters are not part of the evaluated policy checkpoint.