RL DOES NOT ALWAYS FORGET LESS: UNDERSTANDING RETENTION THROUGH TARGET LEARNING IN LLM POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) often preserves prior capabilities better than supervised fine-tuning (SFT) during post-training, but we show that this retention advantage is not universal and can reverse. We study this variation by connecting target learning to non-target retention through gradient analysis and controlled experiments across diverse models, target tasks, and learning rate settings. We first decompose the SFT gradient into a capability component that increases correct-response probability and a supervision component that fits the distribution over provided correct responses. Under binary rewards, the expected on-policy RL gradient is proportional only to the capability component. This distinction reveals how the initial model affects target learning differently under RL and SFT. For RL, low initial correct-response probabilities can limit reward variation and restrict target learning at small learning rates. For SFT, compatible supervision can enable comparable target gains at smaller learning rates and, in some settings, better retention than RL. We further show that the learning rate regime is closely tied to non-target retention: at small learning rates, degradation is typically localized and cross-entropy changes are largely explained by first-order gradient update interactions, whereas at larger learning rates, nonlinear contributions can become substantial and degradation can extend across more non-target tasks. Together, these results explain when RL's retention advantage holds or breaks down and identify the roles of the initial model, target learning efficiency, and learning rate regime in the relative retention of RL and SFT.

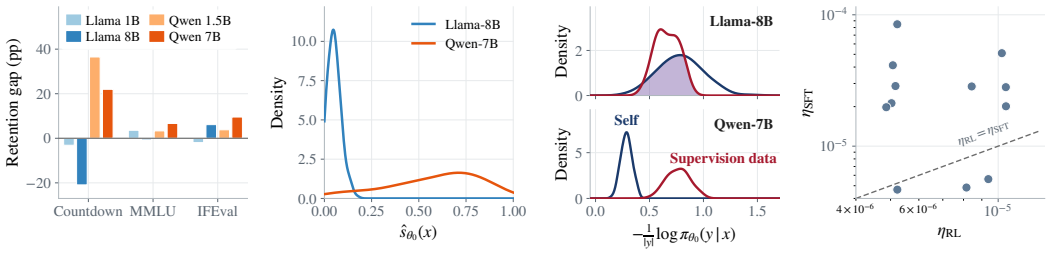
1 INTRODUCTION

Post-training plays a central role in adapting pretrained large language models (LLMs) to target tasks. Supervised fine-tuning (SFT) and reinforcement learning (RL) are two widely used paradigms for such adaptation (Ouyang et al., 2022; Lambert et al., 2025b; Guo et al., 2025). However, adaptation to a target task can affect capabilities acquired during pretraining. When such changes lead to degradation in previously acquired capabilities, the model exhibits catastrophic forgetting, which remains a major challenge in post-training (McCloskey & Cohen, 1989; Kirkpatrick et al., 2017).

Recent studies have shown that, when achieving comparable performance on a target task, RL can cause less degradation of previously acquired model capabilities than SFT, exhibiting a retention advantage over SFT (Chen et al., 2025; Shenfeld et al., 2025; Lai et al., 2026; Chu et al., 2025). This advantage has been attributed in part to RL's use of on-policy data (Chen et al., 2025; Shenfeld et al., 2025). However, it remains unclear why the two post-training paradigms differ in their effects on prior capabilities, which factors govern the magnitude of this retention gap, and when the advantage of RL disappears or reverses. This motivates two questions: what determines the retention gap between RL and SFT, and under what conditions does the advantage of RL break down?

To address these questions, we first examine how SFT and RL acquire target capabilities, and then analyze how their parameter updates affect prior capabilities. We study 12 model–target-task combinations across four base models and three target tasks, evaluate retention on six non-target tasks, and sweep 9 SFT and 12 RL learning-rate settings. Comparing runs at matched target accuracy gains reveals the following:

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.



(a) RL v.s. SFT *(b)* Correct-response probability *(c)* Self vs. supervision data *(d)* Learning rate at matched gain

Figure 1: **RL-SFT differences in target learning and non-target retention depend on the initial model and learning-rate regime.** **(a)** Retention gap: RL's average non-target accuracy change minus SFT's at matched target accuracy gain; positive values favor RL.The x-axis denotes the target task used for post-training. **(b)** Estimated initial correct-response probability $\hat{s}_{\theta_0}(x)$ on Countdown. **(c)** Per-token NLL under θ_0 of the model's own correct responses and SFT supervision responses on Countdown. **(d)** Learning rates required by RL and SFT to reach matched target accuracy gains.

Finding 1: The retention advantage of RL over SFT is not universal. RL often preserves prior capabilities better than SFT while achieving comparable target accuracy gains (i.e., improvements in target-task accuracy). However, our experiments demonstrate that this advantage is not universal (Figure 1(a)): the retention gap of RL and SFT varies across target tasks, models, and optimization settings, and the advantage of RL can disappear under certain training conditions.

Finding 2: Target-learning efficiency depends on the initial model. We characterize initial target capability by the correct-response probability $s_{\theta_0}(x)$, the probability that the initial model θ_0 generates a correct response to prompt x over the target task. **For RL,** as shown in Figure 1(b), this probability can vary substantially across base models. When it is low, on-policy samples provide limited reward variation, restricting RL target learning at small learning rates. **For SFT,** its target-learning efficiency instead depends on how compatible the supervision is with the initial model's distribution over provided correct responses. Figure 1(c) shows substantial variation in this compatibility across models. Consequently, SFT can sometimes reach a matched target accuracy gain with a smaller learning rate than RL (Figure 1(d)) and retain prior capabilities better (Figure 1(a)).

Finding 3: Larger learning rates shift retention degradation from selective to widespread. At **small learning rates**, retention remains high on most non-target tasks, with degradation concentrated on a small subset. As shown in Figure 4, this selective degradation is largely captured by first-order interactions between evaluation gradients and training updates: tasks whose gradients conflict more strongly with the update exhibit poorer retention. At **larger learning rates**, larger target-task gains often come with greater retention costs (Figure 5). In this regime, nonlinear effects become more pronounced, and degradation spreads across a broader set of non-target tasks.

Together, these findings explain why RL does not always retain better than SFT: the outcome depends on the initial model, how efficiently each method acquires the target capability, and the learning-rate regime required to achieve comparable target gains. Beyond these empirical findings, our main **technical contributions** are as follows:

- We conduct a systematic comparison of RL and SFT at comparable target accuracy gains across four base models, three target tasks, and broad learning-rate regimes, providing a controlled empirical framework for studying their relative retention.
- We establish a gradient-level connection between target learning and retention. We decompose the SFT gradient into a capability component that increases correct-response probability and a supervision component that fits the distribution over correct responses. Under binary rewards, the expected on-policy RL gradient follows only the capability component, revealing how the initial model and supervision affect target learning.
- We develop a retention-dynamics analysis that decomposes changes in cross-entropy on responses correctly generated by the base model into first-order gradient-update interactions and nonlinear contributions. This decomposition provides a principled way to characterize when retention is dominated by gradient alignment and when nonlinear effects become substantial.

2 Related Work

Catastrophic Forgetting. Catastrophic forgetting refers to the degradation of previously acquired knowledge or capabilities when a model learns new information (McCloskey & Cohen, 1989; Kirkpatrick et al., 2017). Classical approaches mitigate forgetting by constraining parameter changes, preserving previous outputs, or replaying past data (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Lopez-Paz & Ranzato, 2017). In large language models, fine-tuning can impair domain knowledge, reasoning, and reading comprehension (Luo et al., 2025), and can also weaken safety alignment(Qi et al., 2024). Recent work has therefore examined whether such performance degradation reflects a genuine loss of prior capabilities. Kotha et al. (2024) argue that fine-tuning can bias the model toward the task distribution seen during training, and show that some seemingly forgotten capabilities can be recovered through prompting. Similarly, Zheng et al. (2025) identify spurious forgetting, where performance drops arise from disrupted task alignment rather than loss of underlying knowledge. Other work relates the severity of forgetting to properties of optimization, such as the flatness of the loss landscape (Li et al., 2024).

A Comparison of Forgetting under SFT and RL. Recent studies have shown differences in forgetting and generalization between SFT and RL. Chu et al. (2025) show that SFT tends to memorize training examples, whereas RL generalizes better to unseen textual and visual variants. Consistent findings are reported for knowledge retention in multimodal models (Lai et al., 2026; Zhang et al., 2025). Lai et al. (2026) explain this advantage by the scaling of gradients according to rewards, whereas Zhang et al. (2025) attribute it to the distribution of training data, as rollouts sampled from the current policy are better aligned with the base model. In language models, both Chen et al. (2025) and Shenfeld et al. (2025) show that RL preserves prior capabilities better than SFT at comparable target task performance. Chen et al. (2025) interpret this difference through the forward KL objective of SFT and the reverse KL objective of RL, linking the latter to mode seeking behavior under multimodal distributions. Shenfeld et al. (2025) instead characterize on policy RL as selecting KL minimal solutions relative to the initial policy among those achieving the target task. Khan et al. (2026) instead explain SFT forgetting through the growing mismatch between static supervision and the evolving model policy. These studies show that RL preserves prior capabilities better than SFT, but the extent of this advantage varies across tasks and training conditions. Our work investigates when the retention advantage of RL over SFT holds or breaks down by connecting target learning, initialization, supervision distribution, and the learning rate required to achieve a given target improvement.

3 An Analytical Framework for Target Learning and Retention

Understanding the RL–SFT retention gap requires connecting target learning to its effects on prior capabilities. We first decompose the target-learning gradients to distinguish the learning signals used by RL and SFT. We then analyze how the resulting parameter updates change the likelihood of previously correct responses, separating gradient–update alignment from nonlinear contributions. Together, these analyses provide a framework for interpreting the empirical results in Section 4.

3.1 Understanding Target Learning through Gradient Decomposition

We consider an autoregressive language model π_θ and post-train it on a target task using either supervised fine-tuning (SFT) or reinforcement learning (RL). Let $\mathcal{D}$ denote the target task input distribution. We denote the initial and post-trained parameters by θ_0 and θ_T.

SFT. For each input x, let $Q^+(\cdot \mid x)$ denote a fixed supervision distribution supported on correct responses. SFT minimizes

$$L_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D},\, y\sim Q^+(\cdot|x)} \left[\log \pi_\theta(y \mid x)\right]. \tag{1}$$

On-policy RL. Let $\pi_{\bar{\theta}}$ denote the current behavior policy. Policy-gradient methods optimize a local surrogate of the form

$$L_{\mathrm{RL}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_{\bar{\theta}}(\cdot|x)} \left[A(x,y) \log \pi_\theta(y \mid x)\right], \tag{2}$$

where $A(x, y)$ is a reward-dependent weight computed from trajectories sampled by $\pi_{\bar{\theta}}$. Our experiments use GRPO; the training protocol and KL-regularized exceptions are described in Section 4.1.

The following proposition makes explicit the objective-level distinction between SFT and on-policy RL with binary rewards.

Proposition 3.1. *For a target input x, let $Y \sim \pi_\theta(\cdot \mid x)$ and define the set of correct trajectories by $\mathcal{S}(x) = \{y : R(x, y) = 1\}$. Let $s_\theta(x) := \Pr_{Y\sim\pi_\theta(\cdot|x)}\big[R(x, Y) = 1\big]$ denote the correct-response probability, and define $q_\theta^+(y \mid x) = \Pr_\theta(Y = y \mid R(x, Y) = 1, x)$. Assume that $0 < s_\theta(x) < 1$ and that the fixed SFT supervision distribution $Q^+(\cdot \mid x)$ satisfies $\operatorname{supp} Q^+(\cdot \mid x) \subseteq \mathcal{S}(x)$. Define $g_{\mathrm{cap}}(\theta; x) := \nabla_\theta[-\log s_\theta(x)]$ and $g_{\mathrm{sup}}(\theta; x) := \nabla_\theta D_{\mathrm{KL}}\big(Q^+(\cdot \mid x)\|q_\theta^+(\cdot \mid x)\big)$. Then the SFT gradient satisfies*

$$g_{\mathrm{SFT}}(\theta; x) = g_{\mathrm{cap}}(\theta; x) + g_{\mathrm{sup}}(\theta; x). \tag{3}$$

For binary-reward on-policy RL, let $g_{\mathsf{P}}^{\mathrm{RL}}(\theta; x)$ denote the expected gradient of the RL loss associated with $\mathsf{P} \in \{\mathrm{Reinforce}, \mathrm{GRPO}\}$. Then

$$g_{\mathsf{P}}^{\mathrm{RL}}(\theta; x) = \alpha_{\mathsf{P}}(\theta; x)\, g_{\mathrm{cap}}(\theta; x), \qquad \alpha_{\mathsf{P}}(\theta; x) > 0. \tag{4}$$

For Reinforce*, $\alpha_{\mathrm{Reinforce}}(\theta; x) = s_\theta(x)$. For GRPO with group size M, let $K \sim \mathrm{Binomial}(M, s_\theta(x))$ denote the number of correct responses in a sampled group, and let σ_K denote the within-group standard deviation of the binary rewards. Then*

$$\alpha_{\mathrm{GRPO}}(\theta; x) = \rho_M(\theta; x)\mathbb{E}_K\left[\frac{K(M-K)}{M^2\sigma_K(1-s_\theta(x))}\,\middle|\, 1 \le K \le M-1\right], \tag{5}$$

where $\rho_M(\theta; x) := \Pr(1 \le K \le M-1) = 1 - s_\theta(x)^M - \big(1-s_\theta(x)\big)^M$.

Remark 3.1. Proposition 3.1 compares the gradients of SFT and binary-reward RL. The capability gradient concerns the total probability assigned to correct trajectories, whereas the supervision gradient arises from matching the conditional distribution over correct trajectories to Q^+. The expected RL gradient follows the capability direction but is scaled by $\alpha_{\mathsf{P}}(\theta; x)$. For GRPO, this scale depends on the correct-response probability $s_\theta(x)$ and the group size M(Tajwar et al., 2026; Yang et al., 2026).

Corollary 3.1. *For an input x and a parameter point θ, suppose $g_{\mathrm{cap}}(\theta; x) \neq 0$. Let $\Delta\theta_{\mathrm{SFT}} = -\eta_{\mathrm{SFT}} g_{\mathrm{SFT}}(\theta; x)$ and $\Delta\theta_{\mathrm{RL}} = -\eta_{\mathrm{RL}} g_{\mathrm{RL}}(\theta; x)$, where $g_{\mathrm{SFT}}(\theta; x) = g_{\mathrm{cap}}(\theta; x) + g_{\mathrm{sup}}(\theta; x)$ and $g_{\mathrm{RL}}(\theta; x) = \alpha(\theta; x) g_{\mathrm{cap}}(\theta; x)$ with $\alpha(\theta; x) > 0$. Suppose the two updates achieve the same positive first-order decrease $\delta > 0$ in capability loss defined as $-\log s_\theta(x)$. Then we have*

$$\frac{\eta_{\mathrm{SFT}}}{\eta_{\mathrm{RL}}} = \frac{\alpha(\theta; x)}{1 + \dfrac{\|g_{\mathrm{sup}}(\theta; x)\|_2}{\|g_{\mathrm{cap}}(\theta; x)\|_2}\cos\big(g_{\mathrm{cap}}(\theta; x),\, g_{\mathrm{sup}}(\theta; x)\big)}, \frac{\|\Delta\theta_{\mathrm{SFT}}\|_2}{\|\Delta\theta_{\mathrm{RL}}\|_2} = \frac{1}{\cos(g_{\mathrm{SFT}}(\theta; x), g_{\mathrm{cap}}(\theta; x))}. \tag{6}$$

Remark 3.2. For SFT, the learning rate required to achieve a given capability improvement depends on how strongly the supervision component reinforces the capability direction relative to the RL coefficient α. Thus, SFT can perform well at small learning rates when the supervision gradient is well aligned with g_{cap}, while RL can require a smaller learning rate when α is sufficiently large.

3.2 From Parameter Updates to Retention

Target-learning gradients alone do not determine retention. We next analyze actual parameter updates through their effect on previously correct responses.

For each evaluation dataset $\mathcal{D}_j$, we collect the prompts for which the base model generates a correct response. Let $\{(x_i^{(j)}, y_i^{(j)})\}_{i=1}^{n_j}$ denote the resulting prompt and response pairs. We define the empirical cross-entropy loss on these pairs as

$$\widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta) = -\frac{1}{n_j}\sum_{i=1}^{n_j}\log \pi_\theta\left(y_i^{(j)} \mid x_i^{(j)}\right). \tag{7}$$

We measure the change in this loss relative to initialization as

$$\widehat{F}_j^{\mathrm{CE}}(T) = \widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_T) - \widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_0) = \sum_{t=0}^{T-1} \left[\widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_{t+1}) - \widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_t) \right]. \tag{8}$$

This provides a differentiable measure of changes in previously correct predictions. A positive value indicates that post-training reduces the likelihood assigned to responses that were correct under the base model. To understand how retention changes during post-training, we decompose the change in CE induced by each parameter update. Let $\Delta\theta_t = \theta_{t+1} - \theta_t$. Assuming that $\widehat{\mathcal{L}}_j^{\mathrm{CE}}$ is twice continuously differentiable on a neighborhood containing the line segment $\{\theta_t + \tau\Delta\theta_t : \tau \in [0,1]\}$, Taylor's theorem with integral remainder gives

$$\widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_{t+1}) - \widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_t) = \underbrace{\nabla_\theta \widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_t)^\top \Delta\theta_t}_{\widehat{I}_{j,t}\text{: gradient–update alignment}} + \underbrace{\int_0^1 (1-\tau)\, \Delta\theta_t^\top \nabla_\theta^2 \widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_t + \tau\Delta\theta_t)\Delta\theta_t \, d\tau}_{\widehat{C}_{j,t}\text{: curvature contribution}}. \tag{9}$$

Here, $\widehat{I}_{j,t}$ captures the first order interaction between the evaluation gradient and the parameter update, while $\widehat{C}_{j,t}$ captures the nonlinear contribution along the update direction. Summing over training steps gives $\widehat{F}_j^{\mathrm{CE}}(T) = \sum_{t=0}^{T-1} \widehat{I}_{j,t} + \sum_{t=0}^{T-1} \widehat{C}_{j,t}$.

This identity applies to either training objective and complements task accuracy with a fixed-response CE diagnostic; CE changes need not imply equivalent accuracy changes.

4 Empirical Analysis of the RL–SFT Retention Gap

In this section, we compare retention at comparable target gains, analyze its training dynamics, and relate RL and SFT behavior to their learning signals at initialization.

4.1 Experimental Setup and Comparison Protocol

We consider three target tasks: IFEval (Zhou et al., 2023; Lambert et al., 2025a) for instruction following, MMLU (Hendrycks et al., 2021a) for general knowledge, and Countdown (Pan et al., 2025) for arithmetic reasoning. We evaluate forgetting on the remaining target tasks together with MATH (Hendrycks et al., 2021b), WildJailbreak (Jiang et al., 2024), and WildGuardTest (Han et al., 2024). Accuracy changes are measured relative to initialization in percentage points; the CE diagnostic in Section 3.2 is used to analyze trajectories. We train the models Llama-3.2-1B-Instruct and Llama-3.1-8B-Instruct (Grattafiori et al., 2024), as well as Qwen-2.5-1.5B-Instruct and Qwen-2.5-7B-Instruct (Yang et al., 2024) with SFT and RL across a broad range of learning rates to capture different training regimes. Dataset construction, data splits, learning-rate grids, training hyperparameters, and evaluation details are provided in Appendix B.4. **For each model and target task, RL and SFT use the same set of training prompts and the same number of optimizer steps.** [1]

Comparison protocol. For each model and target task, we select one RL run and one SFT run. The pair is considered gain matched if their target accuracy gains differ by at most 2 percentage points. Otherwise, we include the pair only if one method has both a larger target accuracy gain and a smaller average non-target accuracy decrease.

KL-regularized exceptions. We use RL without KL regularization by default. If training without KL collapses(the target task accuracy decreases), we use KL regularization with $\beta_{\mathrm{KL}} = 0.05$. We mark these KL regularized runs with $*$.

4.2 Target Learning and Retention at Comparable Gains

Retention at comparable target gains. Proposition 3.1 establishes an objective-level distinction between SFT and RL. We now study how this distinction affects target learning and retention in practice. This local, first-order comparison does not establish equivalence between complete training

[1] Our code is available at https://anonymous.4open.science/r/forgetting-rl-sft-A344/.

Table 1: Target learning and retention comparison between RL and SFT under the protocol in Section 4.1. Target accuracy gains and signed average non-target changes are reported in percentage points. Negative average changes indicate forgetting. Bold marks the method with better average retention.

		Learning rate		target accuracy gain		Avg. non-target change	
Target	Model	RL	SFT	RL	SFT	RL	SFT
Countdown	Llama 3.2 1B	8e-6	5e-6	+22.60	+23.40	-1.48	**+1.64**
	Llama 3.1 8B	1e-5*	6e-6	+39.50	+39.50	-20.61	**+0.30**
	Qwen 2.5 1.5B	5e-6	8e-5	+47.10	+40.10	**+0.70**	-35.78
	Qwen 2.5 7B	5e-6	4e-5	+39.00	+19.00	**-0.12**	-22.09
MMLU	Llama 3.2 1B	5e-6	2e-5	+7.98	+7.79	**-0.14**	-3.72
	Llama 3.1 8B	5e-6*	5e-6	+7.15	+11.26	-0.57	**+0.10**
	Qwen 2.5 1.5B	1e-5	3e-5	+4.21	+4.26	**-1.18**	-4.49
	Qwen 2.5 7B	5e-6	3e-5	+9.79	+10.19	**+1.05**	-5.59
IFEval	Llama 3.2 1B	1e-5	2e-5	+25.26	+24.22	-7.06	**-5.13**
	Llama 3.1 8B	5e-6	2e-5	+29.83	+23.25	**-7.36**	-13.48
	Qwen 2.5 1.5B	9e-6	3e-5	+30.91	+32.29	**+0.65**	-3.15
	Qwen 2.5 7B	1e-5	5e-5	+25.33	+24.10	**+0.37**	-9.16

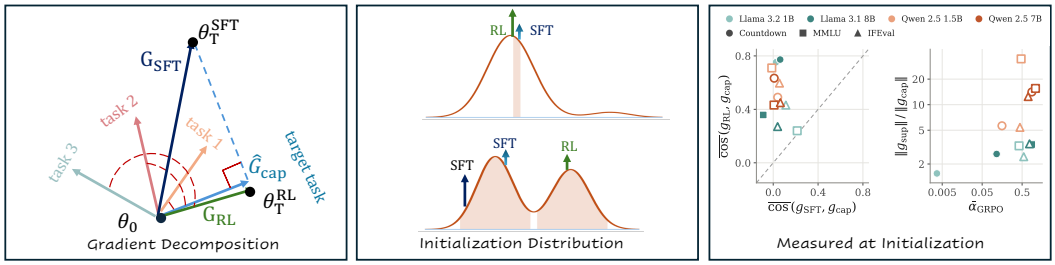
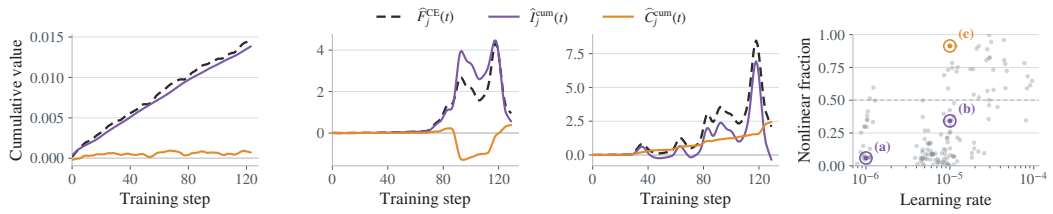


Figure 2: Schematic illustration and measurement of target learning under SFT and RL. **Left:** The expected RL gradient is proportional to g_{cap}, while the SFT gradient adds a supervision component. **Middle:** Response distributions at initialization; shaded regions are correct responses. **Right:** Gradient quantities measured at initialization across all model–task settings. Filled markers: SFT needs a learning rate no larger than RL's to reach the matched gain.

runs. We sweep the learning rate for both SFT and RL across all models and target tasks. As shown in Table 1, across most model target pairs, RL reaches the matched target accuracy gain at a smaller learning rate than SFT. **But this advantage is not universal.** The main exception is Countdown on the Llama family: for both Llama 3.2 1B and Llama 3.1 8B, SFT reaches the same gain at a smaller learning rate than RL. SFT also retains better for Llama 3.1 8B on MMLU and Llama 3.2 1B on IFEval. These reversals indicate that the choice of training objective alone does not determine which method retains better. It also depends on the base model and target task.

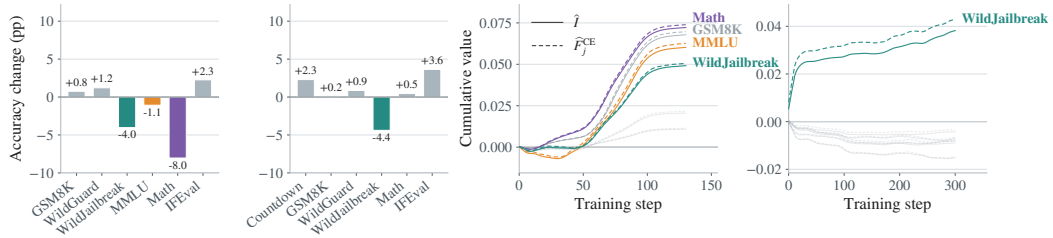
Interpreting the comparison through target-learning gradients. Under binary rewards, the expected on-policy RL gradient contains only αg_{cap}, while SFT adds g_{sup} to fit supervision within correct responses (Proposition 3.1). At initialization, RL is better aligned with g_{cap} in all twelve settings (Figure 2, right).

For equal first-order capability improvement, Corollary 3.1 gives an RL step norm no larger than SFT's. SFT's supervision component can increase movement and alter alignment with non-target gradients (Figure 2, left), offering a local explanation for RL's common retention advantage consistent with prior work (Chen et al., 2025; Shenfeld et al., 2025). The learning-rate ratio depends on α and the supervision component along g_{cap}, linking target learning to initial correct-response probability and supervision compatibility. In our measurements, SFT reaches the matched gain at a learning rate no larger than RL's only when both $\bar{\alpha}$ and $\|g_{\text{sup}}\|_2/\|g_{\text{cap}}\|_2$ are small, as for Llama



(a) Small LR (b) Large LR, first-order (c) Large LR, nonlinear (d) Across settings

Figure 3: **Retention dynamics across learning-rate regimes.** Panels (a)–(c) show the cumulative CE change and its first-order and nonlinear contributions over training. Panel (a) evaluates MMLU after Llama 3.2 1B is trained on IFEval at η=10^{-6}. Panel (b) evaluates WildJailbreak and panel (c) evaluates Math, both after RL training on Countdown of Llama 3.1 8B at η=10^{-5}. Panel (d) compares the relative magnitudes of the cumulative first-order and nonlinear contributions at the final step. The highlighted markers correspond to the cases in panels (a) to (c).



(a) Llama 3.2 1B | Countdown (b) Llama 3.1 8B | MMLU (c) Retention dynamics | Countdown (d) Retention dynamics | MMLU

Figure 4: Selective forgetting under small learning rates. Panels (a,b) show non-target accuracy changes after training Llama 3.2 1B on Countdown and Llama 3.1 8B on MMLU, respectively. Panels (c,d) show the corresponding batch-token-normalized cumulative CE change $\widehat{F}_j^{\mathrm{CE}}$ (dashed) and first-order contribution $\widehat{I}$ (solid). Forgotten datasets are highlighted with matching colors; all others are shown in gray.

on Countdown (right, filled markers). The middle panel illustrates the underlying initial response distributions.

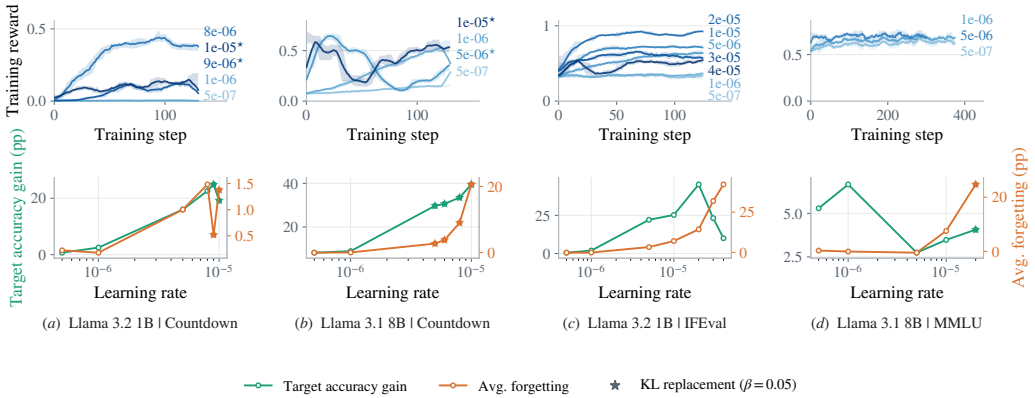
We next examine retention along training, then the learning signals available at initialization.

4.3 Retention Dynamics across Learning-Rate Regimes

We use the CE decomposition in Section 3.2 to examine retention dynamics and task-specific alignment.

Alignment and nonlinear contributions. In Figure 3(a), the first-order contribution closely tracks the CE loss change at the small learning rate. At the larger learning rate, it remains dominant on WildJailbreak (Figure 3(b)). For the same training run, the nonlinear contribution accounts for the final CE loss increase on Math (Figure 3(c)). Across settings in Figure 3(d), nonlinear contributions are more often dominant at larger learning rates.

Selective Forgetting under Small Learning Rates. We first analyze RL runs in which forgetting is concentrated on a few evaluation tasks. In Figure 4(a), Llama 3.2 1B has the largest accuracy drops on Math and WildJailbreak after Countdown training. For Llama 3.1 8B trained on MMLU, the largest drop occurs on WildJailbreak (Figure 4(b)). Performance on the remaining tasks stays relatively stable or improves. To understand these differences, Figure 4(c,d) compare the cumulative CE loss change $\widehat{F}_j^{\mathrm{CE}}$ with its first-order contribution. The tasks with the largest accuracy drops also have substantial CE loss increases. Most datasets plotted in gray have smaller or negative cumulative CE loss changes. In both panels, the cumulative first-order contribution closely tracks $\widehat{F}_j^{\mathrm{CE}}$.



(a) Llama 3.2 1B | Countdown (b) Llama 3.1 8B | Countdown (c) Llama 3.2 1B | IFEval (d) Llama 3.1 8B | MMLU

Figure 5: **RL training dynamics and the learning-rate trade-off, in the four settings where RL retains worse than SFT at matched target accuracy gain.** Top: mean training reward over training steps, with each curve corresponding to a different learning rate. Under binary rewards, the mean reward is the fraction of sampled responses that are correct in each training batch. Bottom: target-task accuracy gain (left axis) and average non-target accuracy decrease (right axis) across learning rates. Larger positive values on the right axis indicate more forgetting.

When the nonlinear contribution is small, the cumulative CE change on evaluation dataset $\mathcal{D}_j$ can be approximated by

$$\widehat{F}_j^{\mathrm{CE}}(T) \approx \sum_{t=0}^{T-1} \left\langle \nabla_\theta \widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_t), \Delta\theta_t \right\rangle = \sum_{t=0}^{T-1} \left\| \nabla_\theta \widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_t) \right\|_2 \|\Delta\theta_t\|_2 \cos\left(\nabla_\theta \widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_t), \Delta\theta_t \right). \tag{10}$$

Since all evaluation datasets share the same parameter updates, differences in the first-order contribution come from the norm of each dataset's evaluation gradient and its alignment with the update in Equation (10). However, an increase in CE loss does not necessarily imply a decrease in task accuracy. GSM8K has a substantial CE loss increase after Countdown training but a small accuracy gain. Improvements on initially incorrect examples offset degradations on initially correct examples. We analyze this case in Appendix C.1.

4.4 Initialization, Target Learning, and Retention

We now relate the learning rates used for comparable target gains to initial reward information in RL and supervision compatibility in SFT.

RL: Limited reward information and the cost of larger learning rates. For RL, limited reward variation can constrain learning at small rates, motivating the comparison between target gains and retention costs below.

Figure 5 examines learning rates up to those selected in Table 1. On Countdown, small rates yield limited gains, while larger gains accompany greater forgetting. With KL regularization at $\eta = 10^{-5}$, Llama 3.1 8B matches SFT's 39.50-point target gain but loses 20.61 points in average non-target accuracy. Under binary rewards, GRPO groups have nonzero advantages only when they contain both correct and incorrect responses. For M independent responses, such an *informative* group has probability

$$\rho_M(\theta; x) = 1 - s_\theta(x)^M - \big(1 - s_\theta(x)\big)^M, \tag{11}$$

Here $s_\theta(x)$ determines the expected reward-gradient coefficient in $g_{\mathrm{RL}} = \alpha_{\mathrm{GRPO}} g_{\mathrm{cap}}$ (Proposition 3.1); alignment with g_{cap} therefore need not imply frequent informative groups. Using 128 initial responses per Countdown prompt and training group size $M = 5$, we estimate mean $(\rho_5, \alpha_{\mathrm{GRPO}})$ as $(0.07, 0.01)$ for Llama 3.2 1B, $(0.25, 0.10)$ for Llama 3.1 8B, and $(0.71, 1.03)$ for

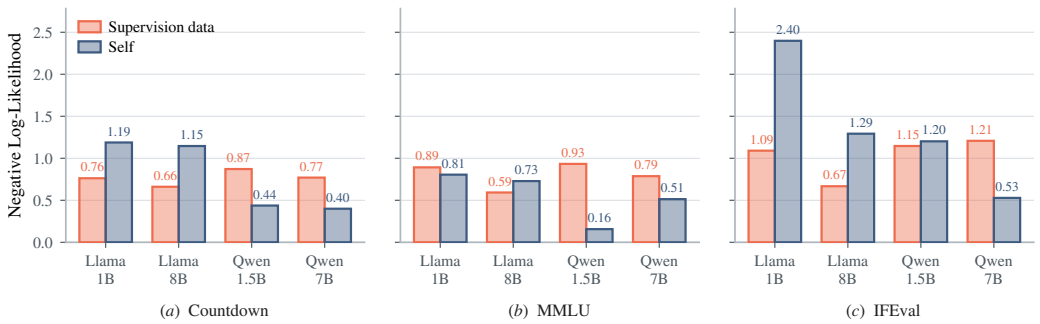


Figure 6: **SFT compatibility and the NLL of the model's own correct responses.** Bars show the average per-token negative log-likelihood under θ_0 of the SFT supervision (ℓ_{sup}) and of that model's own correct responses from 128 rollouts (ℓ^+_{self}).

Qwen 2.5 7B (Appendix C.2). Thus, early Llama learning relies on few informative groups, explaining limited gains at small rates. Larger rates can reach comparable SFT gains but amplify harmful alignment and nonlinear costs.

SFT: Compatible supervision and retention at small learning rates. We now turn from limited reward information in RL to supervision compatibility in SFT, asking when SFT learns effectively at small learning rates.

SFT achieves near-zero or positive average non-target accuracy changes for both Llama models on Countdown and Llama 3.1 8B on MMLU (Table 1). These cases use learning rates below 10^{-5}, whereas most others require larger rates. This agrees with Lin et al. (2026), who relate strong target gains with little forgetting at small rates to initial training-token predictions. In our decomposition, $g_{\text{sup}}(\theta; x)$ increases the supervision response's probability within the distribution over correct responses (Proposition 3.1), motivating a compatibility diagnostic.

Figure 6 compares initial per-token NLL for supervision (ℓ_{sup}) and the model's own correct responses (ℓ^+_{self}). The condition $\ell_{\text{sup}} \leq \ell^+_{\text{self}}$ indicates that supervision is predicted at least as well, providing evidence of compatibility. The well-retaining SFT cases satisfy this condition.

On the same target datasets, both Qwen models instead have $\ell_{\text{sup}} > \ell^+_{\text{self}}$ and worse retention than SFT. The initial model's predictions of its own correct responses also matter: across models, ℓ^+_{self} is generally higher on IFEval than on Countdown or MMLU, as is ℓ_{sup}. Both supervision compatibility and initial correct-response predictability thus appear relevant to SFT retention.

Supervision likelihood indicates compatibility, but does not directly measure g_{sup} or guarantee retention.

5 Conclusion and Limitation

Our work sheds light on how initialization and target learning shape the relative retention of RL and SFT. Limited reward information can restrict RL learning at small learning rates. Larger learning rates can help RL reach target gains comparable to SFT but may also increase forgetting across non-target tasks. Our loss decomposition links this retention cost to first-order interactions and nonlinear contributions. SFT can retain well with compatible supervision at small learning rates. Its retention also depends on how well the initial model predicts its own correct responses.

Limitation. Group size M and sampling temperature T also influence target learning under GRPO. Our experiments keep both settings fixed. Their effects on target learning and retention provide directions for future research.

AI USE STATEMENT

We used generative AI tools to assist with idea discussion, experimental design, literature searches, and manuscript polish. The authors reviewed and revised the AI-assisted content and take full responsibility for the correctness and integrity of the final work.

ETHICS STATEMENT

This work studies forgetting during large language model post-training using existing models and benchmark datasets. Safety-related evaluations involve potentially harmful prompts and are conducted to assess changes in model behavior after post-training.

REPRODUCIBILITY STATEMENT

The model checkpoints, datasets, training configurations, and evaluation protocols are described in Appendix B.4. An anonymized implementation, including training and evaluation scripts, is provided.

REFERENCES

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting, December 2025.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. In Forty-Second International Conference on Machine Learning, June 2025.

Aaron Grattafiori et al. The llama 3 herd of models. arXiv, 2024. URL https://arxiv.org/abs/2407.21783.

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

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. In Advances in Neural Information Processing Systems 37 (NeurIPS 2024) — Datasets & Benchmarks Track, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021a. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021b. URL https://arxiv.org/abs/2103.03874.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In Advances in Neural Information Processing Systems 37 (NeurIPS 2024) — Datasets & Benchmarks Track, 2024.

Rana Muhammad Shahroz Khan, Zijie Liu, Zhen Tan, Charles Fleming, and Tianlong Chen. Tms: Trajectory-mixed supervision for reward-free, on-policy sft, February 2026.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13):3521–3526, 2017.

Suhas Kotha, Jacob Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In International Conference on Learning Representations, volume 2024, pp. 24110–24139, 2024.

Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Qingfu Zhang, Hongbin Liu, Gaofeng Meng, and Fei Zhu. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training, January 2026.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025a. URL https://arxiv.org/abs/2411.15124.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In Second Conference on Language Modeling, 2025b. URL https://openreview.net/forum?id=i1uGbfHHpH.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. In Findings of the association for computational linguistics: EMNLP 2024, pp. 4297–4308, 2024.

Zhizhong Li and Derek Hoiem. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12):2935–2947, 2017.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, Yufan Guo, Kai Zhong, Weiqi Zhang, sujay sanghavi, Changyou Chen, Hyokun Yun, and Lihong Li. SFT doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in LLMs. In The Fourteenth International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=ru93xpQFi1.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. Advances in neural information processing systems, 30, 2017.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025. URL https://arxiv.org/abs/2308.08747.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In Psychology of learning and motivation, volume 24, pp. 109–165. Elsevier, 1989.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35: 27730–27744, 2022.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025. Accessed: 2025-01-24.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In International Conference on Learning Representations, volume 2024, pp. 30988–31043, 2024.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In The Fourteenth International Conference on Learning Representations, October 2025.

Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning. In Forty-third International Conference on Machine Learning, 2026. URL https://openreview.net/forum?id=EeuLO2BjFN.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. arXiv, 2024. URL https://arxiv.org/abs/2412.15115.

Ningyuan Yang, Weihua Du, Weiwei Sun, Sean Welleck, and Yiming Yang. Gradalign: Gradient-aligned data selection for LLM reinforcement learning. In Third Conference on Language Modeling, 2026. URL https://openreview.net/forum?id=bGUY6CJSbu.

Zhihao Zhang, Qiaole Dong, Qi Zhang, Enyu Zhou, Jun Zhao, Zhiheng Xi, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Mingqi Wu, Yanwei Fu, Tao Ji, Tao Gui, Xuanjing Huang, and Kai Chen. Why reinforcement fine-tuning enables mllms preserve prior knowledge better: A data perspective. In The Fourteenth International Conference on Learning Representations, October 2025.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. In International Conference on Learning Representations, volume 2025, pp. 67081–67146, 2025.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A NOTATION

Table 2: Summary of notation used in the paper.

Symbol	Description
General setup	
$\theta,\ \theta_0,\ \theta_t,\ \theta_T$	Model parameters; pretrained, step t, and final parameters
π_θ	Autoregressive policy parameterized by θ
$\bar{\pi}_\theta$	Current behavior policy used to sample on-policy rollouts; treated as fixed when differentiating the local RL surrogate
$\mathcal{D}$	Target task input distribution
$\{\mathcal{D}_j\}_{j\in\mathcal{J}}$	Non target evaluation datasets and their index set
$x,\ Y,\ y$	Input prompt, random trajectory, and realized trajectory
$Q^+(\cdot \mid x)$	Fixed SFT supervision distribution over correct responses
$R(x,y)\in\{0,1\}$	Binary reward or correctness of trajectory y on input x
$S(x)$	Set of correct trajectories, $S(x)=\{y: R(x,y)=1\}$
$s_\theta(x)$	Correct-response probability, $\Pr_{Y\sim\pi_\theta(\cdot\|x)}[R(x,Y)=1]$
$q_\theta^+(y \mid x)$	Policy distribution conditioned on correctness, $\Pr_\theta(Y=y \mid R(x,Y)=1,x)$
$\eta,\ \eta_t$	Constant or step dependent learning rate
$\Delta\theta_t$	Realized parameter update, $\theta_{t+1}-\theta_t$
D_P	Endpoint parameter distance, $\|\theta_T-\theta_0\|_2$
Objectives and gradients	
$\mathcal{L}_{\mathrm{SFT}}(\theta)$	SFT objective
$\mathcal{L}_{\mathrm{SFT}}(\theta;x)$	Prompt level SFT loss
$\mathcal{L}_{\mathrm{RL}}(\theta)$	Local RL surrogate with rollouts sampled from $\bar{\pi}_\theta$
$A(x,y)$	Reward dependent weight in the RL surrogate
$\mathcal{L}_{\mathrm{cap}}(\theta;x)$	Capability loss, $-\log s_\theta(x)$
$g_{\mathrm{cap}}(x)$	Capability gradient, $\nabla_\theta[-\log s_\theta(x)]$
$g_{\mathrm{sup}}(x)$	Supervision gradient, $\nabla_\theta D_{\mathrm{KL}}\left(Q^+(\cdot \mid x)\|q_\theta^+(\cdot \mid x)\right)$
$g_{\mathrm{SFT}}(x)$	SFT gradient, $g_{\mathrm{cap}}(x)+g_{\mathrm{sup}}(x)$
$g_{\mathsf{P}}^{\mathrm{RL}}(\theta;x)$	Expected reward driven RL gradient for algorithm $\mathsf{P}\in\{\mathsf{RF},\mathsf{GRPO}\}$
$\alpha_{\mathsf{P}}(x)$	Positive scalar satisfying $g_{\mathsf{P}}^{\mathrm{RL}}(\theta;x)=\alpha_{\mathsf{P}}(x)g_{\mathrm{cap}}(x)$
ϑ	Auxiliary optimization variable used in the Appendix proof of the local RL surrogate
Retention dynamics	
$(x_i^{(j)},y_i^{(j)})$	Evaluation prompt and corresponding response correctly generated by the base model
n_j	Number of retained evaluation examples in $\mathcal{D}_j$
$\widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta)$	Empirical cross entropy loss on base model correct responses of evaluation dataset $\mathcal{D}_j$
$\widehat{F}_j^{\mathrm{CE}}(T)$	Cumulative CE change, $\widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_T)-\widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_0)$
$\widehat{I}_{j,t}$	First order gradient update interaction, $\nabla_\theta\widehat{\mathcal{L}}_j^{\mathrm{CE}}(\theta_t)^\top\Delta\theta_t$
$\widehat{C}_{j,t}$	Nonlinear contribution along the parameter update segment
τ	Integration variable in the exact Taylor remainder
GRPO and informative group availability	
M	Number of rollouts per prompt during GRPO training
$Y^{(m)},\ R^{(m)}$	mth sampled trajectory in a GRPO group and its binary reward
K	Number of successful trajectories in a group, $K\sim\mathrm{Binomial}(M,s_\theta(x))$
σ_K	Within group reward standard deviation
$\widehat{A}^{(m)}$	Normalized GRPO advantage of rollout m
$A_+(K),\ A_-(K)$	Advantage values for successful and unsuccessful trajectories conditioned on K

Table 2 – continued

Symbol	Description
$\mu_\theta^+(x),\ \mu_\theta^-(x)$	Conditional expected score for successful and failed trajectories
$\rho_M(x)$	Probability that a GRPO group contains both successful and unsuccessful responses
G	Number of rollouts used to estimate the base policy success probability
$\widehat{s}_{\theta_0}(x)$	Empirical success rate of the base policy estimated from G rollouts
$\bar{\rho}_5$	Average informative group probability across prompts at the training group size $M = 5$
β_{KL}	KL regularization coefficient used in KL regularized RL runs
Supervision compatibility	
$\mathcal{X}_{\mathrm{sup}}$	Target prompts used for the supervision compatibility analysis
$y^{\mathrm{sup}}(x)$	SFT supervision response for prompt x
n_x^+	Number of successful base model responses retained for prompt x
$y_{x,k}^+$	kth successful base model response for prompt x
$\bar{\ell}_{\theta_0}(y \mid x)$	Mean per token NLL of response y under the base model, $-\lvert y\rvert^{-1}\log \pi_{\theta_0}(y \mid x)$
ℓ_{sup}	Mean per token NLL of the SFT supervision
ℓ_{self}^+	Mean per token NLL of the base model's own successful responses
Δ_{sup}	Supervision compatibility gap, $\ell_{\mathrm{sup}} - \ell_{\mathrm{self}}^+$

B TARGET TASK LEARNING AND RETENTION UNDER RL AND SFT

B.1 UNDERSTANDING TARGET LEARNING THROUGH GRADIENT DECOMPOSITION

B.1.1 PROOF OF PROPOSITION 3.1

We prove the SFT and RL statements separately. Throughout the proof, we fix a target input x and suppress the conditioning on x whenever no ambiguity arises. We assume $0 < p_\theta(x) < 1$, so that both the correct and wrong conditional trajectory distributions are well defined.

SFT. Let $Z := R(x, Y) \in \{0, 1\}$ denote the binary correctness variable. By definition, $s_\theta(x) = \Pr_\theta(Z = 1 \mid x)$. For any $y \in \mathcal{S}(x)$, the event $\{Y = y\}$ implies $\{Z = 1\}$ since $R(x, y) = 1$. Hence $\{Y = y, Z = 1\} = \{Y = y\}$, and by the definition of conditional probability,

$$q_\theta^+(y \mid x) = \Pr_\theta(Y = y \mid Z = 1, x) = \frac{\Pr_\theta(Y = y, Z = 1 \mid x)}{\Pr_\theta(Z = 1 \mid x)} = \frac{\pi_\theta(y \mid x)}{s_\theta(x)}.$$

Therefore, every correct trajectory satisfies

$$\pi_\theta(y \mid x) = s_\theta(x)\, q_\theta^+(y \mid x), \qquad y \in \mathcal{S}(x). \tag{12}$$

Since $\operatorname{supp} Q^+(\cdot \mid x) \subseteq \mathcal{S}(x)$, Equation (12) holds for every trajectory in the support of the supervision distribution. Under the softmax policy considered here, these trajectories have positive model probability. Hence,

$$\begin{aligned} L_{\mathrm{SFT}}(\theta; x) &= -\mathbb{E}_{y\sim Q^+(\cdot|x)}\left[\log \pi_\theta(y \mid x)\right] \\ &= -\mathbb{E}_{y\sim Q^+(\cdot|x)}\left[\log s_\theta(x) + \log q_\theta^+(y \mid x)\right] \\ &= -\log s_\theta(x) - \mathbb{E}_{y\sim Q^+(\cdot|x)}\left[\log q_\theta^+(y \mid x)\right]. \end{aligned} \tag{13}$$

For brevity, write $Q^+ = Q^+(\cdot \mid x)$ and $q_\theta^+ = q_\theta^+(\cdot \mid x)$. Using the standard cross-entropy decomposition,

$$-\mathbb{E}_{y\sim Q^+} \log q_\theta^+(y) = H(Q^+) + D_{\mathrm{KL}}\left(Q^+ \| q_\theta^+\right), \tag{14}$$

we obtain

$$L_{\mathrm{SFT}}(\theta; x) = -\log s_\theta(x) + D_{\mathrm{KL}}\left(Q^+ \| q_\theta^+\right) + H(Q^+). \tag{15}$$

The supervision distribution Q^+ is fixed, so $H(Q^+)$ is independent of θ. Differentiating Equation (15) gives

$$\begin{aligned}\nabla_\theta L_{\mathrm{SFT}}(\theta; x) &= \nabla_\theta[-\log s_\theta(x)] + \nabla_\theta D_{\mathrm{KL}}\left(Q^+ \| q_\theta^+\right) \\ &= g_{\mathrm{cap}}(x) + g_{\mathrm{sup}}(x),\end{aligned} \tag{16}$$

which proves Equation (3).

The following argument for the RL part follows the analysis in Yang et al. (2026).

On-policy RL. For the RL objectives, the trajectories are sampled from the current policy π_θ. To distinguish the current behavior policy from the optimization variable of the local surrogate, we introduce an auxiliary parameter ϑ inside the proof. The on-policy gradient is obtained by differentiating the local surrogate with respect to ϑ and then evaluating it at $\vartheta = \theta$.

REINFORCE. Consider the local surrogate

$$\mathcal{L}_{\mathsf{RF}}(\vartheta; \theta, x) = -\mathbb{E}_{Y \sim \pi_\theta(\cdot|x)}\left[R(x, Y) \log \pi_\vartheta(Y \mid x)\right]. \tag{17}$$

Treating the sampling distribution as fixed,

$$\begin{aligned}g^{\mathrm{RL}}_{\mathrm{Reinforce}}(\theta; x) &:= \nabla_\vartheta \mathcal{L}_{\mathrm{Reinforce}}(\vartheta; \theta, x)\big|_{\vartheta=\theta} \\ &= -\mathbb{E}_{Y \sim \pi_\theta(\cdot|x)}\left[R(x, Y) \nabla_\theta \log \pi_\theta(Y \mid x)\right] \\ &= -\nabla_\theta s_\theta(x) \\ &= s_\theta(x) g_{\mathrm{cap}}(\theta; x).\end{aligned} \tag{18}$$

Hence

$$\alpha_{\mathrm{Reinforce}}(\theta; x) = s_\theta(x). \tag{19}$$

Sequence level GRPO. For each prompt x, sample

$$Y^{(1)}, \ldots, Y^{(M)} \overset{\text{i.i.d.}}{\sim} \pi_\theta(\cdot \mid x), \tag{20}$$

and let

$$R^{(m)} = R\left(x, Y^{(m)}\right), \qquad K = \sum_{m=1}^{M} R^{(m)}. \tag{21}$$

Then

$$K \sim \mathrm{Binomial}\left(M, s_\theta(x)\right). \tag{22}$$

For a nondegenerate group $1 \le K \le M - 1$, let

$$\widehat{A}^{(m)} = \frac{R^{(m)} - K/M}{\sigma_K}, \tag{23}$$

where $\sigma_K > 0$ is the within group reward standard deviation. For $K = 0$ or $K = M$, we set $\widehat{A}^{(m)} = 0$.

Define the conditional expected scores

$$\mu_\theta^+(x) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(Y \mid x) \mid R(x, Y) = 1\right], \tag{24}$$

and

$$\mu_\theta^-(x) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(Y \mid x) \mid R(x, Y) = 0\right]. \tag{25}$$

Using the score function identity,

$$\mu_\theta^+(x) = \nabla_\theta \log s_\theta(x) = -g_{\mathrm{cap}}(\theta; x), \tag{26}$$

while

$$\mu_\theta^-(x) = \nabla_\theta \log\left(1 - s_\theta(x)\right) = \frac{s_\theta(x)}{1 - s_\theta(x)} g_{\mathrm{cap}}(\theta; x). \tag{27}$$

Therefore,

$$\mu_\theta^+(x) - \mu_\theta^-(x) = -\frac{1}{1 - s_\theta(x)} g_{\text{cap}}(\theta; x). \tag{28}$$

Conditioned on K,

$$\begin{aligned}\mathbb{E}\left[g_{\text{GRPO}}^{\text{RL}}(\theta; x) \mid K\right] &= -\frac{1}{M}\left[K A_+(K)\mu_\theta^+(x) + (M-K)A_-(K)\mu_\theta^-(x)\right] \\ &= \frac{K(M-K)}{M^2\sigma_K\left(1 - s_\theta(x)\right)} g_{\text{cap}}(\theta; x),\end{aligned} \tag{29}$$

$$A_+(K) = \frac{M-K}{M\sigma_K}, \qquad A_-(K) = -\frac{K}{M\sigma_K}. \tag{30}$$

Taking expectation over $K \sim \text{Binomial}(M, s_\theta(x))$ gives

$$g_{\mathsf{GRPO}}^{\text{RL}}(\theta; x) = \alpha_{\mathsf{GRPO}}(\theta; x) g_{\text{cap}}(\theta; x), \tag{31}$$

where

$$\alpha_{\mathsf{GRPO}}(\theta; x) = \mathbb{E}_K\left[\frac{K(M-K)}{M^2\sigma_K\left(1 - s_\theta(x)\right)} \mathbf{1}\{1 \le K \le M-1\}\right]. \tag{32}$$

For $0 < s_\theta(x) < 1$,

$$\Pr(1 \le K \le M-1) = 1 - s_\theta(x)^M - \left(1 - s_\theta(x)\right)^M > 0, \tag{33}$$

and hence

$$\alpha_{\text{GRPO}}(\theta; x) > 0. \tag{34}$$

Combining the REINFORCE and GRPO cases yields

$$g_{\mathsf{P}}^{\text{RL}}(\theta; x) = \alpha_{\mathsf{P}}(\theta; x) g_{\text{cap}}(\theta; x), \qquad \mathsf{P} \in \{\text{Reinforce}, \text{GRPO}\}. \tag{35}$$

B.1.2 Proof for Corollary 3.1

Proof. Since the two updates achieve the same positive first order decrease δ in capability loss,

$$\delta = \eta_{\text{SFT}} \langle g_{\text{cap}}, g_{\text{SFT}} \rangle = \eta_{\text{RL}} \langle g_{\text{cap}}, g_{\text{RL}} \rangle. \tag{36}$$

Using $g_{\text{RL}} = \alpha g_{\text{cap}}$ and $g_{\text{SFT}} = g_{\text{cap}} + g_{\text{sup}}$, we obtain

$$\langle g_{\text{cap}}, g_{\text{RL}} \rangle = \alpha \|g_{\text{cap}}\|_2^2,$$

and

$$\langle g_{\text{cap}}, g_{\text{SFT}} \rangle = \|g_{\text{cap}}\|_2^2 + \langle g_{\text{cap}}, g_{\text{sup}} \rangle.$$

Substituting these identities into (36) gives

$$\frac{\eta_{\text{SFT}}}{\eta_{\text{RL}}} = \frac{\alpha \|g_{\text{cap}}\|_2^2}{\|g_{\text{cap}}\|_2^2 + \langle g_{\text{cap}}, g_{\text{sup}} \rangle}.$$

For the update magnitudes,

$$\frac{\|\Delta\theta_{\text{SFT}}\|_2}{\|\Delta\theta_{\text{RL}}\|_2} = \frac{\eta_{\text{SFT}} \|g_{\text{SFT}}\|_2}{\eta_{\text{RL}} \alpha \|g_{\text{cap}}\|_2}.$$

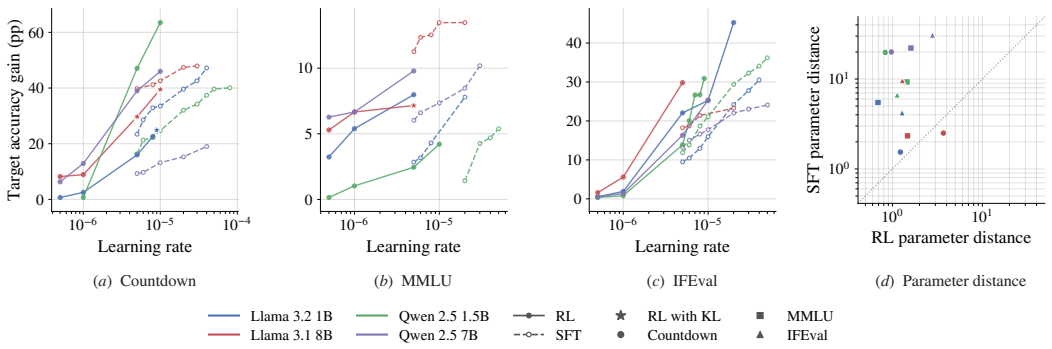
Applying the learning rate ratio above yields

$$\frac{\|\Delta\theta_{\text{SFT}}\|_2}{\|\Delta\theta_{\text{RL}}\|_2} = \frac{\|g_{\text{cap}}\|_2 \|g_{\text{SFT}}\|_2}{\langle g_{\text{cap}}, g_{\text{SFT}} \rangle} = \frac{1}{\cos(g_{\text{SFT}}, g_{\text{cap}})}.$$

Because $\delta > 0$ and $\eta_{\text{SFT}} > 0$, (36) implies $\langle g_{\text{cap}}, g_{\text{SFT}} \rangle > 0$. Hence $0 < \cos(g_{\text{SFT}}, g_{\text{cap}}) \le 1$, and therefore

$$\frac{\|\Delta\theta_{\text{SFT}}\|_2}{\|\Delta\theta_{\text{RL}}\|_2} \ge 1.$$

□



(a) Countdown (b) MMLU (c) IFEval (d) Parameter distance

Figure 7: (a-c) Target accuracy gain (pp) versus learning rate for Countdown, MMLU, and IFEval, for all four base models. (d) Parameter distance $\|\theta_T - \theta_0\|_2$ at the matched gain settings, plotted as RL's(x-axis) against SFT's (y-axis) for each (model, target) pair, both on a log scale.

Table 3: Matched-target-gain RL–SFT comparison with per-dataset accuracy changes. All values are percentage-point changes. The final column averages all six non-target tasks; negative values indicate forgetting. RL* uses KL regularization.

Target	Model	Method	LR	Countdown	GSM8K	WildGuard	WildJailbreak	MMLU	Math	IFEval	Non-target avg.
Countdown	Llama 3.2 1B	RL	8e-6	+22.60	+0.76	+1.22	-4.03	-1.08	-8.04	+2.27	-1.48
	Llama 3.2 1B	SFT	5e-6	+23.40	+1.52	+1.57	+2.67	+2.25	+1.40	+0.41	+1.64
	Llama 3.1 8B	RL*	1e-5	+39.50	-9.40	-7.30	+2.67	-56.22	-34.86	-18.53	-20.61
	Llama 3.1 8B	SFT	6e-6	+39.50	-0.08	+0.87	-0.14	+1.13	+1.36	-1.36	+0.30
	Qwen 2.5 1.5B	RL	5e-6	+47.10	+0.53	-0.29	+2.13	-0.49	+1.12	+1.23	+0.70
	Qwen 2.5 1.5B	SFT	8e-5	+40.10	-62.40	-17.51	-11.90	-54.85	-51.34	-16.68	-35.78
	Qwen 2.5 7B	RL	5e-6	+39.00	+3.64	-1.04	-1.18	-1.81	-0.18	-0.16	-0.12
	Qwen 2.5 7B	SFT	4e-5	+19.00	+3.18	-11.42	-11.67	-64.64	-16.44	-31.58	-22.09
MMLU	Llama 3.2 1B	RL	5e-6	-0.10	+0.00	-0.06	-2.85	+7.98	+1.00	+1.18	-0.14
	Llama 3.2 1B	SFT	2e-5	-1.20	-2.27	+3.88	-12.49	+7.79	-0.14	-10.12	-3.72
	Llama 3.1 8B	RL*	5e-6	+1.30	-0.15	+1.22	-6.61	+7.15	-0.68	+1.49	-0.57
	Llama 3.1 8B	SFT	5e-6	+4.90	-0.08	+0.12	-2.62	+11.26	+0.58	-2.31	+0.10
	Qwen 2.5 1.5B	RL	1e-5	-3.40	-2.05	-0.64	-0.90	+4.21	-0.26	+0.15	-1.18
	Qwen 2.5 1.5B	SFT	3e-5	-7.20	-4.70	+4.93	-4.03	+4.26	-9.16	-6.81	-4.49
	Qwen 2.5 7B	RL	5e-6	+5.10	+0.08	-0.58	-0.63	+9.79	+1.48	+0.82	+1.05
	Qwen 2.5 7B	SFT	3e-5	-6.30	-2.65	-2.43	-3.53	+10.19	-6.38	-12.23	-5.59
IFEval	Llama 3.2 1B	RL	1e-5	-3.70	-3.34	-0.58	-23.35	+1.13	-12.52	+25.26	-7.06
	Llama 3.2 1B	SFT	2e-5	-2.80	+2.20	+0.41	-36.74	+3.92	+2.22	+24.22	-5.13
	Llama 3.1 8B	RL	5e-6	+0.50	-0.61	-2.20	-38.69	+5.34	-8.50	+29.83	-7.36
	Llama 3.1 8B	SFT	2e-5	+4.60	-2.43	-13.04	-74.62	+4.21	+0.38	+23.25	-13.48
	Qwen 2.5 1.5B	RL	9e-6	+1.80	-0.07	-8.18	+10.59	-0.05	-0.20	+30.91	+0.65
	Qwen 2.5 1.5B	SFT	3e-5	-1.50	-4.47	-0.87	-11.27	+2.30	-3.08	+32.29	-3.15
	Qwen 2.5 7B	RL	1e-5	+2.60	+0.83	-1.51	-3.30	+4.41	-0.80	+25.33	+0.37
	Qwen 2.5 7B	SFT	5e-5	-24.10	-5.38	-11.13	-11.27	+2.25	-5.36	+24.10	-9.16

B.2 Target Task Performance under Gain Matched Settings

Figure 7(d) also reports the parameter distance $\|\theta_T - \theta_0\|_2$ at the matched gain settings from Table 1. At these comparable target accuracy gains, RL usually reaches the target with a smaller parameter distance than SFT. Llama 3.1 8B on Countdown is the main counterexample: RL requires a larger parameter distance than SFT to reach the matched gain, mirroring its exceptional behavior in Figure 7(a) and its worse retention at that setting in Table 1.

B.3 Forgetting at Comparable Target Accuracy Gains

Table 3 provides the complete per-dataset results for the matched target accuracy gain comparison, reporting the accuracy change on the target task and each of the six non-target evaluation tasks for every selected RL and SFT pair.

Table 4: Hyperparameters used in our post-training experiments. Braces indicate the values explored across the learning-rate sweeps; not every learning rate was evaluated for every model–task pair. RL denotes group-relative policy optimization with on-policy responses. The main RL experiments use no KL regularization ($\beta = 0$), while $\beta = 0.05$ is included as a controlled variant.

Hyperparameter	**SFT**	**RL**
Base models	Llama 3.2 1B, Llama 3.1 8B, Qwen 2.5 1.5B, Qwen 2.5 7B	
Target tasks	Countdown, MMLU, IFEval	
Evaluation tasks	Countdown, WildGuard, WildJailbreak, MMLU, Math, IFEval	
Learning rate	$\{5, 6, 8\} \times 10^{-6}, \{1, 2, 3, 4, 5, 8\} \times 10^{-5}$	$\{5\} \times 10^{-7}$, $10^{-6}, \{5, 6, 7, 8, 9\} \times 10^{-6}, \{1, 2, 3, 4, 5\} \times 10^{-5}$
Optimizer	AdamW	AdamW
Adam parameters	$\beta_1 = 0.9,\ \beta_2 = 0.95,\ \epsilon = 10^{-8}$	$\beta_1 = 0.9,\ \beta_2 = 0.95,\ \epsilon = 10^{-8}$
LR scheduler	Constant	Constant
Training epochs	2	2
Batch size	64, 128	64, 128
Weight decay	0	0
Precision	bfloat16	bfloat16
RL-only hyperparameters		
Rollouts per prompt	–	5
Rollout temperature	–	1.0
Maximum generated tokens	–	3000
KL coefficient	–	$\beta \in \{0, 0.05\}$

Table 5: Decomposition of the endpoint accuracy change for GSM8K and Math after training Llama 3.2 1B on Countdown. Positive transfer counts examples that change from wrong to correct, while negative transfer counts examples that change from correct to wrong.

Dataset	Positive transfer	Negative transfer	Net accuracy change
GSM8K	$129/1319 = 9.78$ pp	$119/1319 = 9.02$ pp	$+0.76$ pp
Math	$331/5000 = 6.62$ pp	$733/5000 = 14.66$ pp	-8.04 pp

B.4 Experimental Setup

We summarize the post-training setup and hyperparameter ranges used for SFT and RL in Table 4. For the supervision datat of SFT training, we use correct responses generated by Llama-3.3-70B-Instruct;

C When Does RL Forget and When Does SFT Retain?

C.1 RL forgets: Selective forgetting under small learning rates

Discussion on the task: GSM8K Figure 4 reveals an apparent discrepancy between behavioral accuracy and the CE based retention metric for GSM8K. In the Llama 3.2 1B model trained on Countdown, GSM8K exhibits no behavioral forgetting, with the final accuracy increasing slightly by 0.76 percentage points. Nevertheless, its cumulative CE change on responses that were correct under the base model is substantial. These two observations are not contradictory, since they measure different quantities. Behavioral forgetting is defined by the net change in evaluation accuracy over the full dataset. In contrast, $\widehat{F}_j^{\mathrm{CE}}$ is computed only on responses that were correct under the base model and measures how their likelihood changes during training.

To make this distinction explicit, Table 5 decomposes the accuracy change into positive transfer and negative transfer. On GSM8K, 129 out of 1319 examples change from wrong to correct, corresponding to 9.78 percentage points of positive transfer, while 119 examples change from correct to wrong, corresponding to 9.02 percentage points of negative transfer. These two effects nearly cancel, resulting in the small positive net accuracy change of $+0.76$ percentage points. In contrast,

Table 6: Effect of rollout group size on Llama 3.1 8B RL training on Countdown ($\eta = 8 \times 10^{-6}$, $\beta_{\mathrm{KL}} = 0.05$). All entries are accuracy changes relative to the base model, in percentage points. The final column averages the six non-target tasks; negative values indicate forgetting.

Rollouts	Target gain (Countdown)	GSM8K	WildGuard	Wild Jailbreak	MMLU	Math	IFEval	Non-target average
5	+33.50	-10.99	-1.33	+2.13	+0.39	-34.88	-9.50	-9.03
10	+36.50	-9.70	-0.29	-0.41	+1.81	-31.68	-7.45	-7.95

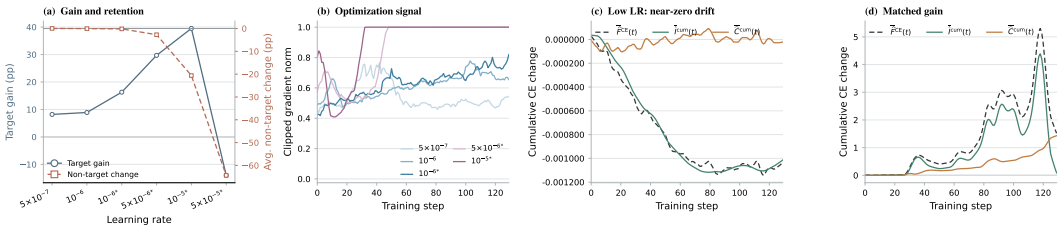


Figure 8: Learning and retention dynamics for RL post-training of Llama 3.1 8B on Countdown. (a) Target-task gain and average non-target accuracy change across learning rates, where * denotes KL regularization. (b) Clipped gradient norm throughout training for the corresponding non-collapsed runs. (c)–(d) Batch-token-normalized cumulative CE change and its decomposition, averaged over the six non-target evaluation datasets.

on Math, negative transfer is substantially larger than positive transfer, leading to a net accuracy change of -8.04 percentage points and hence clear behavioral forgetting.

This example highlights why the CE based quantity is used as a diagnostic of retention dynamics rather than as a direct definition of behavioral forgetting. A task can undergo substantial changes in the likelihood of previously correct responses and considerable example level transitions while showing little or no behavioral forgetting at the aggregate accuracy level.

C.2 RL FORGETS: WIDESPREAD FORGETTING UNDER LARGE LEARNING RATES.

Effect of the rollout group size. Section 4.3 attributes the limited target gains of RL at small learning rates to the scarcity of informative groups. By Eq. (11), a larger group size M raises the probability ρ_M that a group contains both correct and incorrect responses; for Llama 3.1 8B on Countdown at initialization, increasing M from 5 to 10 raises the average ρ_M from about 0.25 to about 0.45. Table 6 tests this at a fixed learning rate ($\eta = 8 \times 10^{-6}$, $\beta_{\mathrm{KL}} = 0.05$). Doubling the group size increases the target accuracy gain from $+33.50$ to $+36.50$ percentage points and slightly reduces the average non-target decrease from -9.03 to -7.95. The forgetting pattern is otherwise unchanged: Math remains the most affected task (-34.88 and -31.68), followed by GSM8K and IFEval. More rollouts per prompt thus provide more reward variation and improve target learning at the same learning rate, but they do not remove the retention cost of RL training in this setting.

The case of RL forgets: Countdown on Llama 3.1 8B Here we provide details of learning and retention dynamics for RL post-training of Llama 3.1 8B on Countdown in Figure 8.

Figure 8 shows how this reduced availability of informative groups is reflected in the subsequent training dynamics of Llama 3.1 8B on Countdown. Panel (a) shows that small learning rates produce little non target degradation but also limited target gains. Increasing the learning rate improves the target task, and the run that reaches a gain comparable to SFT is accompanied by a substantial drop in average non target accuracy. Panel (b) further shows that the clipped gradient norm used in the actual parameter update becomes substantially larger as the learning rate increases. Panels (c,d) further show how these larger parameter updates affect retention. At the small learning rate, the cumulative CE change remains close to zero. At the matched gain learning rate, the CE change becomes much larger, with the nonlinear contribution becoming increasingly important during training. Together,

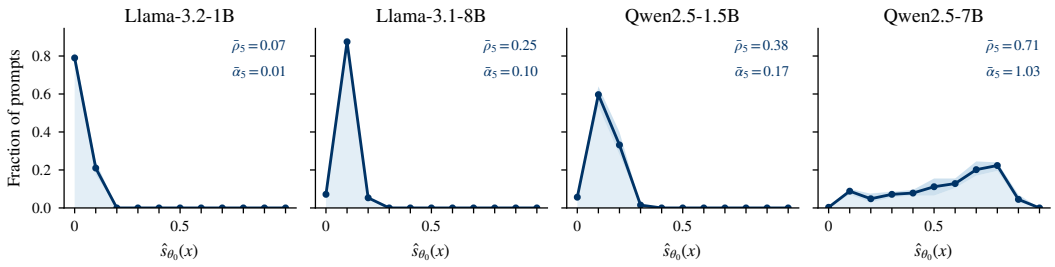


Figure 9: Distribution of $\hat{s}_{\theta_0}(x)$ on Countdown prompts from the RL training pool. Each panel reports $\bar{\rho}_5$ and $\bar{\alpha}_5$.

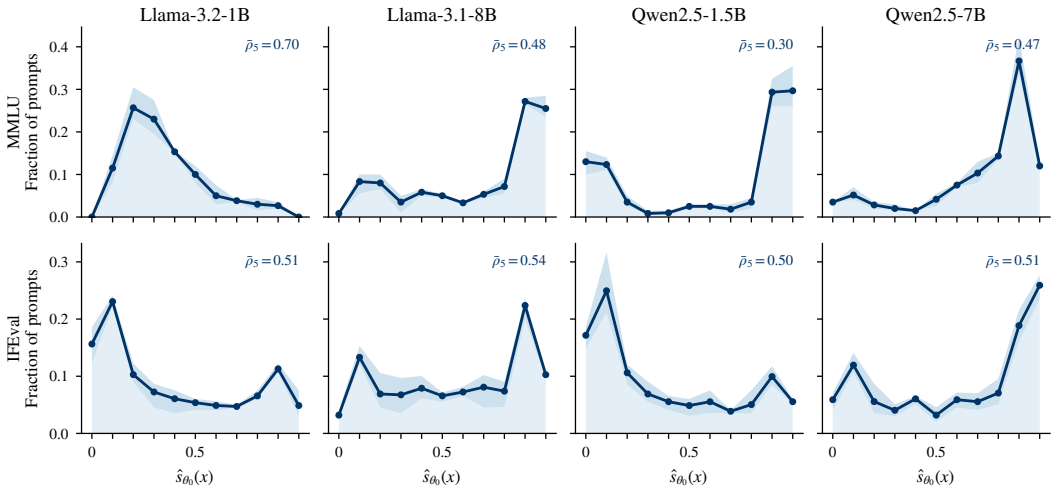


Figure 10: Same as Figure 9 for MMLU (top) and IFEval (bottom); each panel reports $\bar{\rho}_5$.

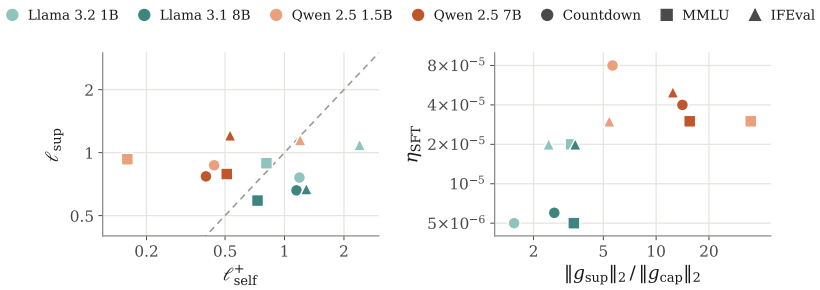
these results suggest that limited availability of informative GRPO groups can make target learning require stronger updates that in turn increase the risk of widespread forgetting.

Correct-response probability at initialization. Figures 9 and 10 show how the correct-response probability of the initial model is distributed across target prompts. For each target task, we sample 200 prompts from the RL training pool and estimate $\hat{s}_{\theta_0}(x)$ from $G = 128$ responses of the initial model per prompt. We bin $\hat{s}_{\theta_0}(x)$ into ten equal-width bins over $(0, 1)$ and plot the fraction of prompts in each bin. We repeat this with 3 independent prompt-sampling seeds: the solid line is the per-bin mean, the light shaded region spans 0 to the per-bin minimum, and the darker region spans the minimum to the maximum across seeds. Each panel reports $\bar{\rho}_5$, the average over prompts of the probability in Eq. (11) that a GRPO group of size $M = 5$ contains both correct and incorrect responses, and, for Countdown, the corresponding average coefficient $\bar{\alpha}_5$ from Eq. (5).

On Countdown (Figure 9), the two Llama models place almost all prompts near $\hat{s}_{\theta_0}(x) = 0$, giving $\bar{\rho}_5 = 0.07$ and 0.25 and $\bar{\alpha}_5 = 0.01$ and 0.10. Most GRPO groups therefore contain no correct response and contribute no gradient, which explains the limited target gains of RL at small learning rates (Section 4.3). Qwen 2.5 1.5B also concentrates at low but non-zero probabilities ($\bar{\rho}_5 = 0.38$), while Qwen 2.5 7B spreads its prompts over the whole interval ($\bar{\rho}_5 = 0.71$, $\bar{\alpha}_5 = 1.03$), so most of its groups are informative.

On MMLU and IFEval (Figure 10), no model has the near-zero concentration of Llama on Countdown; $\bar{\rho}_5$ lies between 0.30 and 0.70. A low $\bar{\rho}_5$ here can also arise from prompts the model already solves: Qwen 2.5 1.5B on MMLU places many prompts near $\hat{s}_{\theta_0}(x) = 1$ as well as near 0, giving the lowest value on this task ($\bar{\rho}_5 = 0.30$) even though its correct-response probability is high on average. Such groups contribute no gradient either, but unlike the Countdown case they do not indicate

limited accessibility of correct responses. On IFEval, all four models have a similar mix of prompts near 0 and near 1, and $\bar{\rho}_5 \approx 0.5$.



(a) Likelihood compatibility *(b) Supervision component vs. η_{SFT}*

Figure 11: **SFT compatibility at initialization.** (a) Per-token negative log-likelihood under θ_0 of the supervision responses (ℓ_{sup}) versus the model's own correct responses (ℓ^+_{self}); points below the diagonal indicate compatible supervision. (b) Median over prompts of $\|g_{\text{sup}}\|_2/\|g_{\text{cap}}\|_2$ versus the SFT learning rate required to reach the matched target accuracy gain (Table 1)

C.3 When does SFT Retain?

Section 4.4 assesses SFT compatibility through the initial model's likelihood of the supervision responses. Figure 11 relates this likelihood-based measure to its gradient-level counterpart in Corollary 3.1. For each prompt in the first SFT training batch, we sample 128 responses from π_{θ_0}, compute $g_{\text{cap}}(x)$ as the average gradient of the sequence-level negative log-likelihood over correct responses and $g_{\text{SFT}}(x)$ as the gradient on the supervision response, and set $g_{\text{sup}} = g_{\text{SFT}} - g_{\text{cap}}$. Prompts without a correct response are excluded, and we report the median of $\|g_{\text{sup}}\|_2/\|g_{\text{cap}}\|_2$ over the remaining prompts, since the ratio is heavy-tailed across prompts.

Panel (a) summarizes Figure 6: the Llama models lie below the diagonal ($\ell_{\text{sup}} < \ell^+_{\text{self}}$) on all three target tasks except MMLU with Llama 3.2 1B, while the Qwen models mostly lie above it. Panel (b) shows that the relative size of the supervision component increases with the learning rate SFT needs to reach the matched target accuracy gain. The settings requiring $\eta_{\text{SFT}} \leq 2 \times 10^{-5}$ have $\|g_{\text{sup}}\|_2/\|g_{\text{cap}}\|_2$ between 1.5 and 3.5, whereas those requiring $\eta_{\text{SFT}} \geq 3 \times 10^{-5}$ have values between 5.4 and 35. In all settings, $\cos(g_{\text{cap}}, g_{\text{sup}}) < 0$ for most prompts, so by Corollary 3.1 a larger supervision component requires a larger SFT learning rate for the same capability improvement.