ANTI-GENERAL INTELLIGENCE: SPECIALIZATION VIA FOCAL FORGETTING AND HYPERPLASTICITY

Anonymous authors
Paper under double-blind review

ABSTRACT

On many ML tasks, frontier performance is achieved by pretrained LLMs that are subsequently specialized to a task by post-training on data from that task. Specialization is typically assumed to come at the cost of forgetting previous abilities, which is welcome in a specialist deployed for a single purpose. However, recent work has shown that post-trained models may retain information even without explicit anti-forgetting measures. Across a wide range of domains, we observe that small post-trained LLMs do indeed retain their pre-specialization performance on non-target tasks, most completely when specialized with RL. We analyze the characteristics of non-forgetting in a saturated specialist, finding that task performance is carried by sparse task-selective subnetworks and that non-target tasks hold capacity the specialist never reclaims. We then study an intervention for the focal forgetting of task information by Low-Rank Ascent, finding that forgetting alone does not reliably help the target on its own, and develop ANTIGEN, a method that improves models beyond saturation by selectively increasing model plasticity in the non-target representation subspaces that ascent identifies. Applied to small specialists, ANTIGEN outperforms direct specialization and contemporary plasticity baselines on math and commonsense reasoning tasks, generalizes better in-domain, and focuses the loss of log-probabilities in ascent-derived subspaces.

1 INTRODUCTION

The advent of large, pretrained foundation models (FMs) has led to a revolution in applied machine learning. At the time of writing, multi-agent systems built from frontier models such as GPT and Claude are the pinnacle of AI problem-solving ability (Gottweis et al., 2026; Novikov et al., 2025; Anthropic, 2025; OpenAI, 2026). Their achievements, however, require a tremendous amount of computation: OpenAI's proposed resolution of the Navier–Stokes Millennium Prize problem required on the order of 10,000 coordinated agents generating roughly 130 billion tokens over 88 hours — equivalent to about a million human-authored books (OpenAI, 2026). For many routine ML tasks, inference with a large model is both unnecessary and economically infeasible. To balance ability and feasibility, practitioners instead rely on smaller *specialized* FMs (Fu et al., 2023; Belcak et al., 2025). Post-training a pretrained FM yields specialists that outperform bespoke ML pipelines in domains as far-ranging as mathematics (Yang et al., 2024; Qu et al., 2026), coding (Roziere et al., 2023; Hui et al., 2024), medicine (Singhal et al., 2023; Zhang et al., 2024a; Vishwanath et al., 2024), law (Colombo et al., 2024; Dominguez-Olmedo et al., 2025), and computational social science (Ziems et al., 2024), while also retaining a natural language interface.

Specialists are deployed in applications whose scope is far narrower than the abilities of the pretrained FM they are built from, whether domain-specific (e.g., mathematics, medicine) or task-specific (e.g., agentic search, scientific writing), so a good specialist should dedicate maximal capacity to its target. We observe, however, that post-trained specialists retain most of their pre-specialization performance on unrelated tasks: those trained with RL are essentially unchanged, and those trained with SFT or distillation show mild to moderate drops with only rare collapses. Capacity dedicated to non-target tasks is therefore largely not repurposed by specialization, and is 'dead weight' in a specialist that is deployed only for its target application.

This non-forgetting is consistent with findings that post-training changes FMs only slightly (Panigrahi et al., 2023; Jain et al., 2024; Zhu et al., 2025; Mukherjee et al., 2026), that apparent forgetting

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.

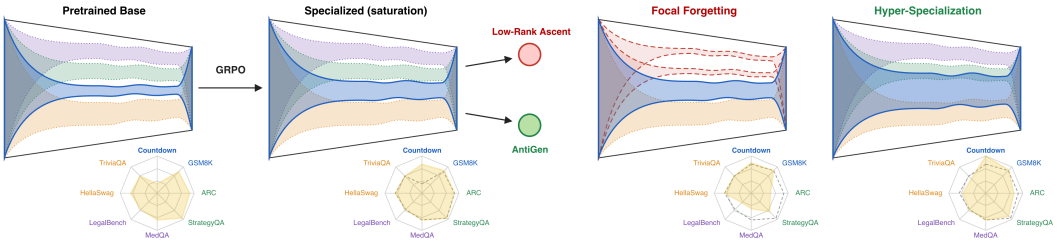


Figure 1: **Specialization, forgetting, and hyper-specialization.** The performance (in the radar plots) of a foundation model (trapezoid) is driven by sparse task-specific subnetworks (colored bands). Specialization (e.g., via GRPO) expands a subnetwork but reclaims little capacity from other tasks, so forgetting is absent or shallow. In this work, we show that (1) Low-Rank Ascent (Section 3.2) can induce focal forgetting of non-target tasks with little effect on the target, and (2) selectively amplifying plasticity with ANTIGEN (Section 3.3) can specialize beyond saturation by directing increases in plasticity and likelihood loss into subspaces seeded by low-rank ascent.

is shallow and recoverable (Davari et al., 2022; Zheng et al., 2025; Inamdar et al., 2026), and with the lottery ticket hypothesis (Frankle & Carbin, 2018; Chen et al., 2020). To further understand this phenomenon, in this paper we study the activations, gradients, and Fisher information of a specialist that is *saturated* on its target task, rather than the smaller networks of prior work, and find the same sparse structure. As a first measure towards reclaiming this capacity, we then directly induce forgetting of non-target tasks, as in machine unlearning (Liu et al., 2025b), with the goal of leaving the target unaffected, and find that forgetting alone helps target performance only in narrow cases. Building on these observations, we develop Anti-General Intelligence (ANTIGEN), which expands the target-task subnetwork by amplifying plasticity in the low-rank subspaces that carry non-target performance (Figure 1). We make the following contributions.

- We demonstrate across domains, training methods, and architectures that small specialized FMs are unable to re-purpose a large fraction of their performance on non-target tasks.
- We characterize the mechanism of non-forgetting in a saturated specialist model and show that focal forgetting can hinder or help target performance.
- We develop ANTIGEN, which post-trains small specialist FMs beyond saturation, outperforming GRPO and contemporary plasticity-restoring baselines on math and commonsense reasoning and yielding reasoning abilities that generalize zero-shot in domain.

2 RELATED WORK

2.1 SPECIALIZED PRETRAINED LANGUAGE MODELS

On many ML tasks, frontier performance is achieved by 'post-training' a foundation model, either by further domain-adaptive pretraining (Gururangan et al., 2020), supervised fine-tuning (SFT) (Ouyang et al., 2022; Chung et al., 2024), distillation (Hinton et al., 2015; Hsieh et al., 2023; Fu et al., 2023), parameter-efficient low-rank adaptation (Houlsby et al., 2019; Hu et al., 2022), or reinforcement learning with proximal or group-relative policy optimization (Shao et al., 2024; Guo et al., 2025). Many practical specialists are below 10 billion parameters in size (Fu et al., 2023; Belcak et al., 2025). Specialization is typically assumed to cause catastrophic forgetting of non-target information, but the literature shows mixed results. In sequential classification tasks, larger pretrained models forget less (Ramasesh et al., 2022), although forgetting under SFT has been reported to grow from 1B to 7B parameters (Luo et al., 2023), potentially due to a shift in the model's task inference (Kotha et al., 2024). Models trained with on-policy RL also appear to forget less than SFT counterparts (Lai et al., 2025) due to a bias towards policies with a minimal KL divergence to the pretrained base (Chen et al., 2025; Shenfeld et al., 2026). Further, RL updates touch only 5–30% of model parameters, though at nearly full rank within that subset (Mukherjee et al., 2026), and avoid the principal directions of the pretrained weights (Zhu et al., 2025). These observations align with the lottery-ticket hypothesis (Frankle & Carbin, 2018; Chen et al., 2020) and with the finding that fine-tuning localizes a skill to $\sim$0.01% of parameters (Panigrahi et al., 2023).

2.2 FORGETTING, RECOVERY, AND UNLEARNING

Limited parameter change is also consistent with forgetting being shallow rather than catastrophic. Fine-tuning (Jain et al., 2024), linear probing (Davari et al., 2022), and even few-shot adaptation (Inamdar et al., 2026) have been shown to recover seemingly forgotten performance after specialization or continual learning, which Zheng et al. (2025) attribute to a recovery of task alignment rather than of knowledge. In some settings, forgetting itself is an explicit goal. The field of *machine unlearning* focuses on the targeted removal of specific information—private records, copyrighted text, hazardous capabilities—while preserving everything else (Cao & Yang, 2015; Liu et al., 2025b; Barez et al., 2025). Methods for LLM unlearning perform gradient ascent on a 'forget dataset' paired with gradient descent on a broad 'retain dataset' (Liu et al., 2022; Jang et al., 2023; Maini et al., 2024), replace unbounded ascent with a preference-style objective (Zhang et al., 2024b), dampen parameters by Fisher importance (Foster et al., 2024), perturb representations (Li et al., 2024), or use unlearning adapters (Chen & Yang, 2023) or task vectors (Ilharco et al., 2022). However, recent work has shown that even information that is explicitly unlearned is not truly forgotten, and can be rapidly recovered by in-context or in-weights relearning, quantization, or representation probes (Shumailov et al., 2024; Deeb & Roger, 2024; Hu et al., 2025; Zhang et al., 2025; Xu et al., 2026).

Our goal is instead to maximize the capacity a model dedicates to its target task. A small body of work pursues this in non-RL settings (Zhou et al., 2022; Chen et al., 2023b). SFT in LLMs can be improved by subtracting parameters corresponding to old knowledge (Ni et al., 2024) or splitting tokens into learning and unlearning sets (Taheri et al., 2025). Closer to our setting, LWF interleaves Fisher-scored gradient ascent on self-generated non-target text with SFT (Jiang et al., 2025), and Exclusive Unlearning pushes such text toward the uniform distribution while fine-tuning on a target (Sasaki et al., 2026). Both act on data rather than parameters, and neither studies a saturated model.

2.3 PLASTICITY AND RETENTION

Our work focuses primarily on increasing plasticity during RL post-training in saturated specialists. Work on plasticity in deep RL and continual learning (Dohare et al., 2024) shows that injecting fresh parameters (Nikishin et al., 2023), resetting dormant ones (Sokar et al., 2023), or perturbing gradient descent based on utility (Elsayed & Mahmood, 2024) can improve plasticity, but these interventions have rarely been studied at LLM scale. At this scale, excessive SFT prior to RL training appears to collapse plasticity (Liu et al., 2026). While these above act on training data and these plasticity interventions on individual parameters, ANTIGEN targets the low-rank subspaces that carry non-target performance. The *opposite strategy*, explicitly retaining information to improve target-task performance, also has support. Replaying pretraining data can improve sample efficiency in later learning (Kotha & Liang, 2026; Marek et al., 2026), and forgetting less predicts better forward transfer in continual learning (Chen et al., 2023a). In RL post-training specifically, prolonged training under a KL penalty with periodic resets of the reference policy to the current policy (Liu et al., 2025a), or constraints that preserve policy entropy (Cui et al., 2025), are used to continue improving past a plateau. We find that neither method improves a saturated specialist in our setting.

3 FOCAL FORGETTING AND HYPERSPECIALIZATION

3.1 FORGETTING IN SMALL SPECIALIST LLMS

Table 1 compares released specialists across domains, scales, and training methods with their base models on a common set of non-target probes. Forgetting is limited across the board: specialists trained with RL are essentially unchanged on non-target tasks, SFT and distillation specialists show mild to significant drops, and catastrophic forgetting is confined to a few cells. Our emphasis is not on any particular setting, but on the fact that *every model shows significant performance on tasks that it will not be deployed to do*. The lack of forgetting in the RL experts is particularly noteworthy given the results in the literature (Section 2), and motivates our experiments in GRPO-based specialization.

3.2 FOCAL FORGETTING WITH LOW-RANK ASCENT

If specialization is independent of non-target performance, a focused intervention should be able to induce forgetting. LoRA (Hu et al., 2022) adds a low-rank update BA to a weight matrix, $h =$

Domain	Base / Specialist	Training	math500	gsm8k	mmlu_stem	mmlu_hum	mmlu_soc	mmlu_other	arc_easy	hellaswag
math	Qwen3-4B-Thinking / QED-Nano	SFT+RL	0.974 / 0.968	0.916 / 0.934	0.706 / 0.710	0.608 / 0.600	0.808 / 0.820	0.755 / 0.763	0.854 / 0.868	0.558 / 0.556
	Qwen3-4B / Polaris-4B-Preview	RL	0.958 / 0.962	0.928 / 0.864	0.724 / 0.714	0.604 / 0.612	0.791 / 0.791	0.738 / 0.738	0.862 / 0.858	0.586 / 0.586
	Qwen3-4B / Pythagoras-Prover-4B	SFT+RL	0.958 / 0.468†	0.928 / 0.616	0.724 / 0.688	0.604 / 0.583	0.791 / 0.775	0.738 / 0.706	0.862 / 0.844	0.586 / 0.558
	Qwen3-4B / Qwen3-4B-Countdown-RLVR	RLVR	0.958 / 0.960	0.928 / 0.928	0.724 / 0.730	0.604 / 0.604	0.791 / 0.792	0.738 / 0.744	0.862 / 0.866	0.586 / 0.590
	Qwen2.5-1.5B / Qwen2.5-Math-1.5B	SFT	0.218 / 0.366	0.388 / 0.476	0.562 / 0.471	0.528 / 0.341	0.717 / 0.498	0.696 / 0.430	0.794 / 0.696	0.602 / 0.510
	Qwen2.5-7B / Qwen2.5-Math-7B	SFT	0.610 / 0.600	0.814 / 0.758	0.720 / 0.655	0.689 / 0.463	0.826 / 0.652	0.791 / 0.617	0.870 / 0.796	0.664 / 0.592
reasoning	Qwen2.5-Math-1.5B / DeepSeek-R1-Distill	Distill	0.366 / 0.830	0.476 / 0.790	0.471 / 0.393	0.341 / 0.303	0.498 / 0.413	0.430 / 0.398	0.696 / 0.688	0.510 / 0.458
	Qwen2.5-Math-7B / DeepSeek-R1-Distill	Distill	0.600 / 0.930	0.758 / 0.886	0.655 / 0.566	0.463 / 0.400	0.652 / 0.615	0.617 / 0.558	0.796 / 0.760	0.592 / 0.540
code	Qwen2.5-1.5B / Qwen2.5-Coder-1.5B	SFT	0.218 / 0.476	0.388 / 0.556	0.562 / 0.531	0.528 / 0.459	0.717 / 0.642	0.696 / 0.613	0.794 / 0.738	0.602 / 0.548
	Qwen2.5-7B / Qwen2.5-Coder-7B	SFT	0.610 / 0.504	0.814 / 0.820	0.720 / 0.669	0.689 / 0.563	0.826 / 0.771	0.791 / 0.751	0.870 / 0.840	0.664 / 0.664
science	Qwen3-4B / Qwen3-4B-SFT-science	SFT	0.958 / 0.750†	0.928 / 0.906	0.724 / 0.716	0.604 / 0.632	0.791 / 0.800	0.738 / 0.730	0.862 / 0.870	0.586 / 0.614
agentic search	Qwen3-4B-Thinking / Jan-v1-4B	RL	0.974 / 0.968	0.916 / 0.930	0.706 / 0.710	0.608 / 0.612	0.808 / 0.808	0.755 / 0.744	0.854 / 0.854	0.558 / 0.576
medical	Qwen3-4B-Thinking / MedPsy-4B	SFT	0.974 / 0.902	0.916 / 0.940	0.706 / 0.710	0.608 / 0.602	0.808 / 0.832	0.755 / 0.751	0.854 / 0.836	0.558 / 0.572
	Llama-3-8B / Llama-3-8B-UltraMedical	SFT	0.036 / 0.176	0.046 / 0.508	0.544 / 0.552	0.587 / 0.598	0.767 / 0.771	0.740 / 0.724	0.834 / 0.846	0.696 / 0.680
legal	Llama-3-8B / Lawma-8B	SFT	0.036 / 0.072	0.046 / 0.060	0.544 / 0.487	0.587 / 0.533	0.767 / 0.632	0.740 / 0.679	0.834 / 0.776	0.696 / 0.644
multilingual	Qwen3-4B-Base / AfriqueQwen-4B	SFT	0.532 / 0.720	0.716 / 0.898	0.779 / 0.734	0.624 / 0.571	0.828 / 0.818	0.781 / 0.730	0.860 / 0.850	0.636 / 0.612

Table 1: Base / specialist accuracy of released specialists on a common set of probes. Specialist scores are colored by change from the base: green zero forgetting or positive transfer, yellow mild ($\leq 5\%$), orange significant ($\leq 15\%$), red catastrophic ($> 15\%$). Forgetting is limited across the board. †Lower bound: $\geq 99\%$ of MATH-500 completions reach the 32k-token budget (Appendix A.2).

$W_0x + BAx$, trains only A and B, and can merge the result into the weights. We instead train the adapter by gradient *ascent* on a forgetting task, so that it spans the directions along which that task's loss is most sensitive, merge it to induce forgetting, and continue training on the target. Assume we are given a set of *donor* tasks $\{\mathcal{T}_i' | i \in [T]\}$ and a specialist model θ_S on a target task $\mathcal{T}$. Then, we do the following steps.

$$B_iA_i \leftarrow B_iA_i^{(0)} + \alpha \sum_{x' \sim \mathcal{T}_i'} \nabla_{BA} \mathcal{L}_i'(W_0 + B_iA_i, x') \qquad \text{(Low-Rank Ascent)} \quad (1)$$

$$W \leftarrow W + \beta \sum_{i \in [T]} B_iA_i, \qquad \text{(merge adapter)} \quad (2)$$

$$\text{Repeat: } W \leftarrow W - \eta \nabla_W \mathcal{L}_t(W, x),\ x \sim \mathcal{T} \qquad \text{(specialize on target)} \quad (3)$$

As we show in Section 4, LRA can induce focal forgetting of non-target tasks in cases where the tasks are not closely aligned with the target. In very few cases, we observe a *positive transfer* to target-task performance beyond continued GRPO updates. However, since we do not explicitly optimize for this transfer, we do not expect this in the general setting. To achieve this, we explicitly optimize the low-rank component in the direction of forgetting, as we describe next.

3.3 Anti-General Intelligence: Specializing Beyond Saturation

As we show in Section 4, even when LRA produces focal forgetting in non-target tasks, specialization does not necessarily benefit from this freed capacity. To transfer this capacity to the target task, we develop a procedure that *selectively increases plasticity* in exactly these subspaces. We collect each A_i into a block matrix $\boldsymbol{A}$ and discard the content matrices B_i, instead re-initializing a matrix of high-plasticity trainable parameters $\hat{\boldsymbol{B}}$. We train these parameters alongside the model weights with the same GRPO optimizer, but at a *higher learning rate* $m \cdot \eta$. Intuitively, the perturbation caused by hyperplasticity in the $k \times T$-dimensional low-rank space corresponding to $\boldsymbol{A}$ should only affect non-target tasks and not conflict with the specialization process, while also aggressively refocusing capacity towards the target.

$$B_iA_i \leftarrow B_iA_i^{(0)} + \alpha \sum_{x \sim \mathcal{T}_i'} \nabla_{BA} \mathcal{L}_i'(W_0 + B_iA_i, x') \qquad \text{(LRA, equation 1)}$$

$$\text{Freeze } \boldsymbol{A}^{Tk \times d} \leftarrow [A_1|A_2|\cdots|A_T]^\top; \quad \text{Re-initialize } \hat{\boldsymbol{B}} \leftarrow \boldsymbol{0}^{h \times Tk} \quad (4)$$

$$\text{Repeat: } [W|\hat{\boldsymbol{B}}] \leftarrow [W|\hat{\boldsymbol{B}}] - \eta \nabla_W \mathcal{L}_t(W + \hat{\boldsymbol{B}}\boldsymbol{A}, x) - m\eta \nabla_{\hat{\boldsymbol{B}}} \mathcal{L}_t(W + \hat{\boldsymbol{B}}\boldsymbol{A}, x),\ x \sim \mathcal{T} \qquad \text{(Specialize } W, \hat{\boldsymbol{B}}) \quad (5)$$

$$W \leftarrow W + \hat{\boldsymbol{B}}\boldsymbol{A} \qquad \text{(merge adapter)} \quad (6)$$

We name this method 'Anti-General Intelligence' (AntiGen) to represent the focus on specialization without regard to the cost to non-target tasks. In our experiments, training small specialist FMs

with ANTIGEN outperforms continued GRPO and other methods for RL plasticity on mathematical and commonsense reasoning tasks (Section 4). We also observe better in-distribution transfer compared to continued GRPO and improved Fisher importance ratios between the target and non-target tasks on ANTIGEN-derived specialists. We describe these and other analyses in Section 5.

4 EXPERIMENTS

4.1 FOCAL FORGETTING WITH LOW-RANK ASCENT

To study whether LRA can produce focal forgetting effects, we first specialize Qwen3.5-0.8B (Qwen Team, 2026), a small but highly capable FM, on a synthetic Countdown task. Modeled on a game show[1], Countdown is an arithmetic search task that requires input numbers to be composed into an arithmetic expression that evaluates to a target number (details in Appendix A.1). Over 1600 steps of GRPO, the accuracy of the model rises from 0% to $\sim 45\%$, and effectively plateaus (with a 5% increase over the next 800 steps). We treat the model at step 1600 as our near-saturated specialist θ_S. In each experiment, we train and merge (with weight $\beta = 0.3$ determined empirically) rank-16 adapters with LRA on a set of donor tasks at the start of continued specialization, and thereafter whenever the training reward plateaus for 20 consecutive steps (Appendix D).

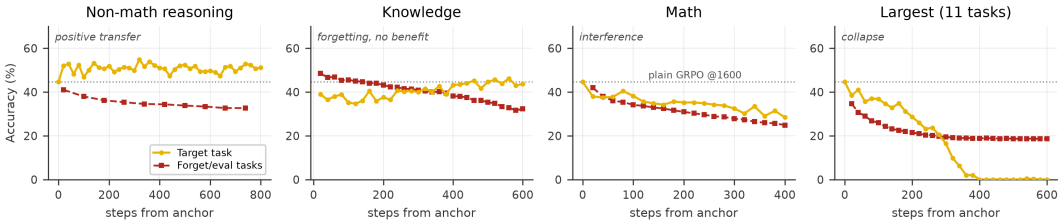


Figure 2: LRA while specializing on Countdown (gold: Countdown; red: donor/eval task mean; *dotted*: 1600-step accuracy $\theta_S = 44.6\%$). Reasoning and knowledge donors forget without harming the target, math donors interfere, and the 11-task suite collapses it.

We group donors as mathematical, non-math reasoning, and knowledge-based according to their interaction with Countdown: forgetting math donors reliably degrades the target, whereas forgetting knowledge or non-math reasoning donors leaves it within noise of the control, with single runs ranging from -4% to $+6\%$ (Figure 2). We also evaluate a range of specialization tasks with the same groupings of donor tasks. Table 2 reports differences against a matched GRPO control for these targets over the same training window. LRA removes donor performance as intended, but collateral damage on evaluation-only *probe* tasks is broad for math donors, and the target is spared only for knowledge and reasoning donors. Forgetting alone therefore does not reliably free capacity for the target, motivating the explicit recruitment of this capacity with ANTIGEN.

	Math			Non-Math Reasoning			Knowledge			Broad (Combined)		
Target Task	target	donor	probe	target	donor	probe	target	donor	probe	target	donor	probe
Countdown	−10.9	−10.3	−13.1	+3.5	−8.2	−4.8	−2.7	−10.8	−8.7	−21.3	−27.6	−34.1
AI2-RC	−4.0	−11.9	−13.8	−4.0	n/m	−11.8	−2.5	−3.3	−0.7	−3.0	−18.4	−2.5
StrategyQA	−0.2	−38.5	−13.3	+0.4	−14.8	−5.3	−0.3	−9.2	−7.7	−1.6	−17.2	−9.2
DeepScaleR	−7.6	−27.3	−9.4	−2.9	−12.9	−5.8	−1.8	−3.2	−2.0	−1.3	−10.5	−3.1
OpenBookQA	−3.1	−12.1	−10.2	−2.6	n/m	−7.1	+0.3	−5.9	−1.9	−4.8	−15.8	−4.6
Lawma (8B, SFT)	+0.2	−0.2	−1.9	+0.1	n/m	−2.3	+0.1	n/m	−0.8	+0.1	−5.8	−0.3
MATH-500	+0.4	−4.7	−6.2	+1.4	−9.9	−5.3	+2.7	−2.8	−7.2	−2.7	−8.6	−12.9

Table 2: LRA by donor family: change vs. the matched control over the same window (points) on the *target* (tgt), the *donor* tasks (fgt), and eval-only *collateral* tasks (coll); n/m = not measured. Target changes of ≥ 2 points are coloured (up, down). Measurement details are presented in Appendix D.

[1] https://en.wikipedia.org/wiki/Countdown_(game_show)

4.2 HYPER-SPECIALIZATION WITH ANTIGEN

Through a comparison with the state of the art on the Countdown task, we show that ANTIGEN outperforms competitors on a saturated specialist. We then replicate our improvements on GRPO on two reasoning tasks: DeepScaleR (Tan et al. (2026)) and OpenBookQA (Mihaylov et al. (2018)).

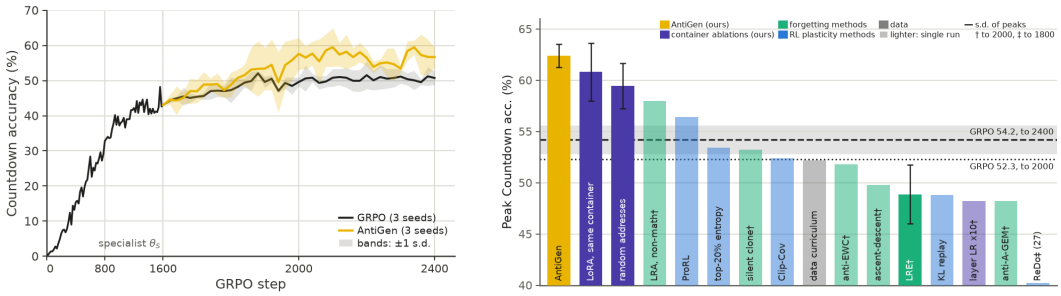


Figure 3: Countdown from θ_S: (a) GRPO vs. ANTIGEN; (b) ANTIGEN vs. ablations and RL plasticity baselines. ANTIGEN beats GRPO throughout training and peaks higher than any other method.

Countdown. Starting from θ_S, we stack the rank-16 projectors of the 21-donor suite into an address matrix of rank 336 and train $\hat{B}$ with a 25× larger effective step ($s = \alpha/336\times25 \approx 2.4$, α=32). Over three seeds, ANTIGEN raises the peak Countdown accuracy from 54.2% to 62.4% and consistently outperforms GRPO over training (Figure 3a). The gain transfers zero-shot to Countdown-3-4 (Pan et al. (2025), peak 65.0% vs. 62.1%), and we observe no degradation in performance on donor tasks. Figure 3b compares ANTIGEN with contemporary remedies for plasticity loss in RL run from the same θ_S under the same budget — ProRL-style reference and optimizer resets (Liu et al., 2025a), Clip-Cov (Cui et al., 2025), high-entropy-token updates (Wang et al., 2026), KL replay, and ReDo (Sokar et al., 2023) — all of which are within noise of GRPO or worse (best +2.2; ReDo −22.1). We present extensive descriptions and a detailed analysis of each ablation in Appendix B.

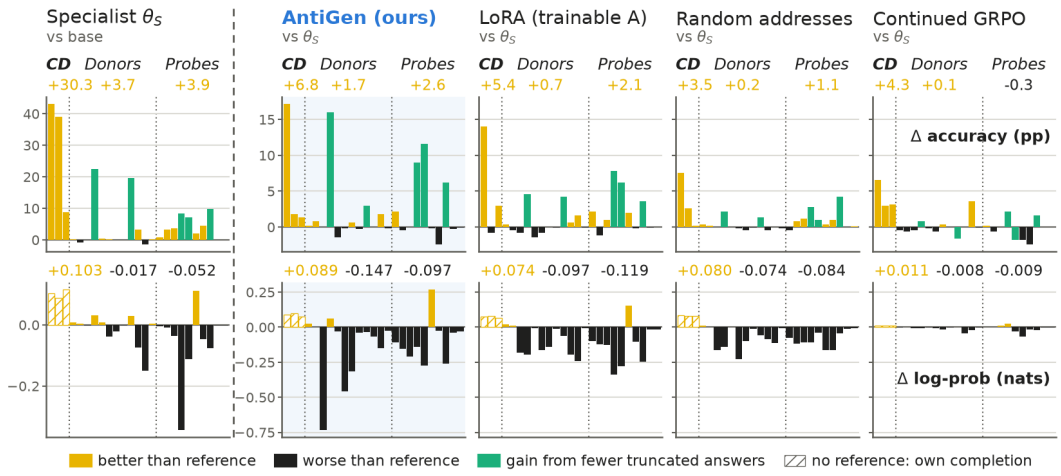


Figure 4: Accuracy (top row) and log-probabilities of teacher-forced outputs (bottom row) of specialization (left column) and hyper-specialization with ANTIGEN, container ablations, and GRPO.

Focal forgetting and reasoning brevity. At first glance, ANTIGEN's performance appears to be comparable to that of high learning rate LoRA or a randomly initialized A matrix (Figure 3b). However, a closer look shows three phenomena that differentiate ANTIGEN from its closest competitors. Figure 4 shows the per-task accuracy and log-probability of teacher-forced solutions across 25 tasks, divided into target, donor, and probe tasks. First, there is naturally some information loss during specialization, as log-probabilities on non-target tasks worsen (bottom left subplot). This effect stops at near-saturation (bottom right, Continued GRPO). Second, discounting the log-probabilities that

are naturally lost by increased accuracy, ANTIGEN shows clearly greater loss of reference log-probability on donor tasks compared to trainable (i.e., LoRA) or frozen random $\boldsymbol{A}$ matrices. This indicates that the LRA-derived frozen $\boldsymbol{A}$ matrices do induce the *focal forgetting* of information, but not necessarily with enough strength to induce significant accuracy loss. Finally, ANTIGEN produces *shorter reasoning chains* than its ablations — both math and non-math reasoning tasks (teal bars) improve in performance simply due to fewer truncations at max reasoning depth, more than other strategies. We present magnified 38-task plots and results on other tasks in Appendix B.

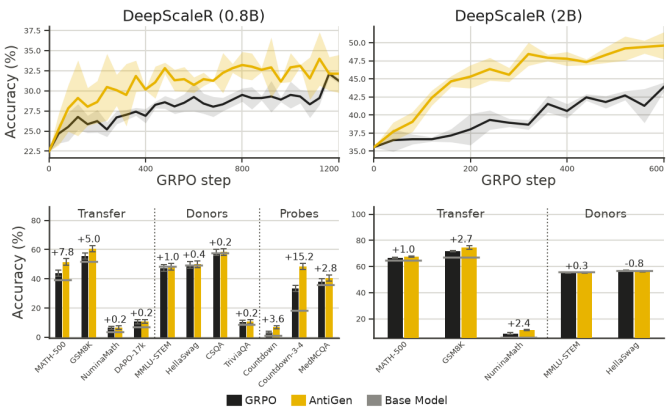


Figure 5: DeepScaleR from base (3 seeds). Top: target accuracy. Bottom: transfer/non-target accuracy at last checkpoint.

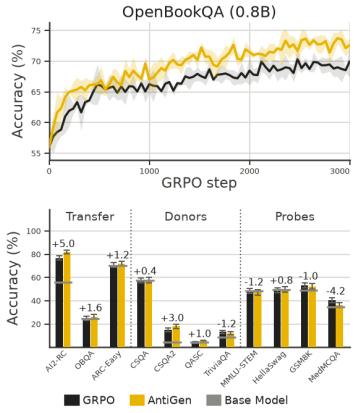


Figure 6: OpenBookQA: specialization, forgetting, transfer.

DeepScaleR. To evaluate a more challenging mathematical setting, we train Qwen3.5-0.8B with GRPO on the DeepScaleR (Tan et al., 2026) dataset of competition-level math problems from AMC, AIME, and other sources (three seeds per arm). ANTIGEN on a broad donor suite not only improves performance on this dataset compared to the GRPO-only control, but also *significantly improves performance on other math benchmarks* — GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021), and retains NuminaMath (LI et al., 2024) and DAPO-17k (Yu et al., 2026) — indicating that the reclaimed capacity produces generally improved mathematical reasoning compared to control (Figure 5). Further, the model does not lose significant non-target (e.g. MMLU-STEM) performance. Since this task is adequately challenging, we replicate the result at more than double the scale: Qwen3.5-2B performs better on DeepScaleR, and generally on math, when trained with ANTIGEN than with plain GRPO. Measuring the log-probabilities shows the same focal forgetting and reduced truncation compared to GRPO, as in the Countdown case (Figure 13; Appendix B).

OpenBookQA and AI2-RC. Having seen improvements in mathematical ability, we now specialize in a different domain: general knowledge and reasoning via the AI2 Reasoning Challenge (AI2-RC; Clark et al., 2018) and OpenBookQA (Mihaylov et al., 2018). The latter is an especially suitable dataset for Qwen3.5-0.8B, as the model starts weak and improves strongly with training. With ANTIGEN on a donor suite selected by Fisher importance ratios and LRA performance on AI2-RC, we see consistently higher performance over the training process than with standard GRPO. Further, this result transfers zero-shot to AI2-RC, and with no observed accuracy loss (Figure 6). Figure 14 in Appendix B shows a similar trend as on DeepScaleR.

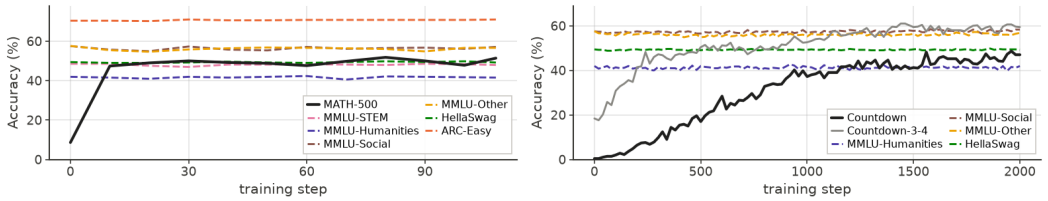
5 ANALYZING NON-FORGETTING AND HYPER-SPECIALIZATION

In this section, we expand on our hypotheses for non-forgetting in specialists, and analyze the mechanism of improvement with ANTIGEN compared to GRPO and competing methods.

5.1 ANALYZING NON-FORGETTING IN A SPECIALIST LLM

To follow the specialization-forgetting curve directly, we specialize Qwen3.5-0.8B on math reasoning with two training methods, measuring in-domain MATH-500 accuracy alongside non-mathematical reasoning probes (Zellers et al., 2019; Clark et al., 2018; Geva et al., 2021a) and

knowledge probes derived from MMLU (Hendrycks et al., 2020). Under supervised fine-tuning on the FineProofs-SFT dataset of the QED-Nano pipeline (Qu et al., 2026), MATH-500 accuracy jumps as the model learns the reasoning strategy and then plateaus, while the non-target probes do not move (Figure 7a). We also train with RLVR using GRPO on our Countdown task. As accuracy rises from 0 to ∼45% over 2000 steps, with a matching rise on the easier Countdown-3-4 set of Pan et al. (2025), the non-target probes again stay flat (Figure 7b). We anchor later experiments at step 1600, where the improvement is approximately $+0.1$%/100 steps from an average of $+2.7$%, and call this θ_S *saturated*. Continued GRPO still gains $+0.7 \pm 0.5$ points / 100 steps over the next 800 steps, and we compare each result in Section 4 against this continuation and not the anchor.



(a) SFT on FineProofs (0.8B, 6 epochs).

(b) GRPO on Countdown (0.8B, 2000 steps).

Figure 7: Specializing Qwen3.5-0.8B: the target (solid) climbs; no non-target forgetting (dashed).

Activation maps. Qwen3.5-0.8B has 24 layers, six repetitions of three Gated DeltaNet layers (Yang et al., 2025) and one full-attention layer, each followed by an MLP. We record the mean activation of every MLP unit of θ_S on data from each task and subtract each unit's cross-task mean, which exposes blocks of task-selective units (Figure 8). Such units are present at every depth: in each layer, 3–7% of MLP units devote more than a quarter of their activation to a single task, while shared units explain 44–94% of activation variance. This is consistent with prior observations of task-selective MLP units in deep networks (Bau et al., 2017; Geva et al., 2021b).

Gradient alignment. The cosine similarity between the average Countdown gradient and that of every other task lies between 0 and 0.06 (Figure 9). All exceed the $1/\sqrt{d} \approx 4 \times 10^{-5}$ expected of unrelated directions in $d = 7.5 \times 10^8$ parameters, indicating a weak shared component and thus a low degree of transfer/interference (Riemer et al., 2019). Self-consistency is high in Countdown batches, with the cosine averaging 0.978 (Appendix C).

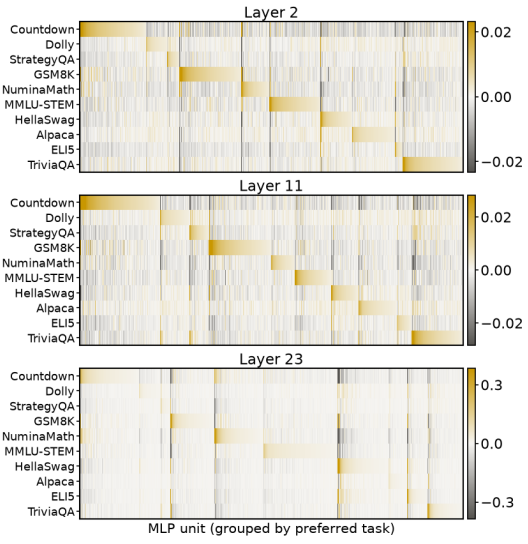


Figure 8: Mean-subtracted MLP activations of θ_S in an early, middle and late layer (columns are units) show task-selective blocks at every depth.

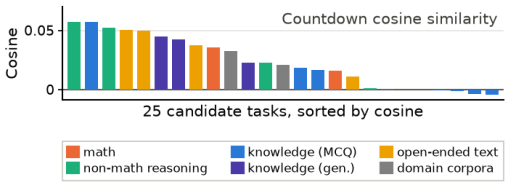


Figure 9: Gradient cosine is small (≤ 0.06).

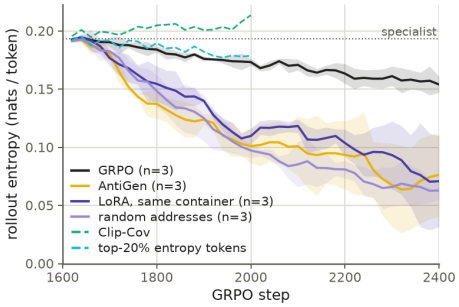


Figure 10: ANTIGEN shows lower rollout entropy than GRPO.

5.2 Analyzing Hyper-specialization with AntiGen

Extensive ablations show that the AntiGen improvement stems from two components. Figure 3 shows that plain LoRA with increased plasticity (+6.6% avg. peak) and AntiGen but with a randomly initialized $\boldsymbol{A}$ matrix (+5.3%) both perform better than any method from the literature, and nearly as well as AntiGen. However, Figure 4 shows that each method sacrifices non-target information in a distinct manner, with only AntiGen producing focal forgetting. Further, AntiGen has lower cross-seed variance, indicating that explicitly selecting donor tasks may provide robustness to random noise. The improvement also cannot be attributed to heightened exploration: adapter strategies show *lower* rollout entropy over training than continued GRPO (0.11–0.12 vs. 0.17 nats/token at 2400 steps, Figure 10), and the two entropy-raising baselines do not beat GRPO.

In addition to the log-probability and accuracy changes (Figure 4), two preliminary observations support the interpretation that AntiGen recruits non-target subspaces. First, AntiGen with only the six knowledge-based donors as addresses, whose forgetting gave LRA no benefit, *still improves* on GRPO (+3.5% at peak in a single 200-step run). Second, the Fisher information ratio of non-target to Countdown tasks lowers with AntiGen but not with continued GRPO (Appendix C.3).

6 Discussion

Our method engages with two broad research directions in the literature (Section 2): shallow, recoverable forgetting in unlearning and continual learning (CL), and plasticity and the lottery ticket hypothesis. Shallow forgetting is increasingly observed in FMs across a range of scales, and our analysis of specialization through the lens of unlearning and CL ties our results to these observations. In contrast with unlearning (where it is highly detrimental) and CL (where it is beneficial), shallow forgetting in specialization is detrimental only in the sense that it occupies capacity that the target may be able to use. Our LRA experiments further show that directed forgetting can be conditionally useful: forgetting tasks that share capacity with the target but do not rely on the same set of skills (e.g., non-mathematical reasoning tasks for Countdown) can transfer positively, whereas other types of tasks may result in no change or even a detriment. Here, we focus on *expanding the capacity* dedicated to the target task for improved performance. An alternative approach is to prune the network of this capacity to make the model smaller and more efficient without affecting target performance (Han et al., 2016; Sanh et al., 2020; Xia et al., 2024).

Analyses of model dynamics, including in RL post-training, show that task- or domain-specific *subnetworks* drive performance on a wide range of tasks in deep neural networks. The literature on plasticity attempts to restore learning ability by explicitly adding capacity to the model or perturbing the training dynamics, and retention-based methods regularize updates toward a reference policy. We find that neither family yields further improvement on an already saturated specialist. AntiGen, which explicitly recruits capacity from non-target subnetworks to increase plasticity on the target, improves the performance of an already-saturated model with GRPO on mathematical and commonsense reasoning tasks. We observe that the ratio of Fisher information between the target and non-target tasks grows after training with AntiGen, compressing toward parity in the late layers, indicating that a larger share of these parameters influences target performance.

Limitations. While we comprehensively evaluate multiple tasks and methods for post-training in a range of settings, our study is primarily limited to the Qwen3.5-0.8B (and 2B) model on reasoning-focused tasks with GRPO. In future work, we aim to study these phenomena on other model families and training methods, and at other scales, e.g., where SFT-induced forgetting is more severe (Luo et al., 2023). While LRA and AntiGen identify and improve upon specialist task performance across settings, the mixture of donor tasks we use to induce forgetting remains empirically determined. Our ablations (particularly with random addresses) and Fisher analysis shed some light on training dynamics, but further work can be done to fully understand the mechanism behind AntiGen's success. Our method also adds a few hyperparameters (LRA merge weight, plasticity scale, and fold interval) to an already complex post-training process, and choosing these empirically adds overhead to specialization. Finally, deriving the $\boldsymbol{A}$ matrix requires additional data and a few steps of low-rank ascent per donor task, and $\hat{\boldsymbol{B}}$ adds parameters during GRPO training (but not inference).

AI USE STATEMENT

In this work,

- We **have used** generative AI tools for the following tasks: proposing and refining hypotheses (specialization techniques and ablations for each technique), designing and providing feedback on research methodology or experiments, implementing methods, cleaning and reformatting datasets, supporting qualitative and thematic data analysis.
- We **have not used** generative AI tools for the following tasks: developing theoretical models or conceptual frameworks, interpreting results.
- The following AI tasks are **not applicable** to our work: Generating synthetic data sets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs.

We have not used generative AI tools for the initial ideation, domain and model selection, and major parts of the implementation of the method. Additionally, we used generative AI tools for the following: creating and modifying scientific figures or images, suggesting experimental parameters, creating and editing software code, drafting parts of a research paper, brainstorming, sourcing/searching for information, editing a research paper to improve readability, and identifying relevant literature.

We have reviewed all AI-assisted work in the following manner.

- All AI-generated code for methodology and ablations was manually reviewed line-by-line by the authors.
- Code used in plotting was manually verified to use the recorded results of experiments to prevent the hallucination of results or information.
- LLM-suggested relevant literature was verified by a manual look up and review of the suggested work, and bibtex entries were retrieved from official sources.
- All LLM-generated writing and editing in this manuscript was subjected to a thorough review and collective editing by the authors.

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 present extensive details on our research methodology and experimental setup in Section 4, and in Appendix F.

REFERENCES

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. Explanations for CommonsenseQA: New Dataset and Models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3050–3065, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.238. URL https://aclanthology.org/2021.acl-long.238/.

Anthropic. How we built our multi-agent research system. https://www.anthropic.com/engineering/multi-agent-research-system, June 2025. Anthropic Engineering Blog. Accessed 2026-09-20.

Fazl Barez, Tingchen Fu, Ameya Prabhu, Stephen Casper, Amartya Sanyal, Adel Bibi, Aidan O'Gara, Robert Kirk, Ben Bucknall, Tim Fist, et al. Open problems in machine unlearning for ai safety. *arXiv preprint arXiv:2501.04952*, 2025.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL https://arxiv.org/abs/1704.05796.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*, 2025.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480, 2015. doi: 10.1109/SP.2015.35.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. *arXiv preprint arXiv:2510.18874*, 2025.

Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12041–12052, 2023.

Jiefeng Chen, Timothy Nguyen, Dilan Gorur, and Arslan Chaudhry. Is forgetting less a good inductive bias for forward transfer? *arXiv preprint arXiv:2303.08207*, 2023a.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained bert networks. *Advances in neural information processing systems*, 33:15834–15846, 2020.

Yihong Chen, Kelly Marchisio, Roberta Raileanu, David Adelani, Pontus Lars Erik Saito Stenetorp, Sebastian Riedel, and Mikel Artetxe. Improving language plasticity via pretraining with active forgetting. *Advances in Neural Information Processing Systems*, 36:31543–31557, 2023b.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of machine learning research*, 25(70):1–53, 2024.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre FT Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, et al. Saullm-7b: A pioneering large language model for law. *arXiv preprint arXiv:2403.03883*, 2024.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

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

MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16691–16700. IEEE, 2022.

Aghyad Deeb and Fabien Roger. Do unlearning methods remove information from language model weights? *arXiv preprint arXiv:2410.08827*, 2024.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, Ashique Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632:768 – 774, 2024. URL https://api.semanticscholar.org/CorpusID:259251905.

Ricardo Dominguez-Olmedo, Vedant Nanda, Rediet Abebe, Stefan Bechtold, Christoph Engel, Jens Frankenreiter, Krishna Gummadi, Moritz Hardt, and Michael Livermore. Lawma: The power of specialization for legal annotation. In *International Conference on Learning Representations*, volume 2025, pp. 35059–35103, 2025.

Mohamed Elsayed and A Rupam Mahmood. Addressing loss of plasticity and catastrophic forgetting in continual learning. In *International Conference on Learning Representations*, volume 2024, pp. 33399–33430, 2024.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 3558–3567, 2019.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12043–12051, 2024.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning*, pp. 10421–10430. PMLR, 2023.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021a.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495. Association for Computational Linguistics, 2021b. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, et al. Accelerating scientific discovery with co-scientist. *Nature*, pp. 1–3, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 8342–8360, 2020.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016. URL https://arxiv.org/abs/1510.00149.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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. *arXiv preprint arXiv:2103.03874*, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the association for computational linguistics: ACL 2023*, pp. 8003–8017, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning. In *International Conference on Learning Representations*, volume 2025, pp. 8857–8888, 2025.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Amogh Inamdar, Matthew So, Vici Milenia, and Richard Zemel. Re-evaluating continual learning with few-shot adaptation. *arXiv preprint arXiv:2606.03843*, 2026.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert Dick, Hidenori Tanaka, Tim Rocktäschel, Edward Grefenstette, and David Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In *International Conference on Learning Representations*, volume 2024, pp. 9944–10015, 2024.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, 2023.

Chunyang Jiang, Chi-min Chan, Yiyang Cai, Yulong Liu, Wei Xue, and Yike Guo. Graceful forgetting in generative language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13176–13191, 2025.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8082–8090, 2020.

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 and Percy Liang. Replaying pre-training data improves fine-tuning. *arXiv preprint arXiv:2603.04964*, 2026.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In *International Conference on Learning Representations (ICLR)*, 2024.

Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Qingfu Zhang, Hongbin Liu, et al. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training. *arXiv preprint arXiv:2507.05386*, 2025.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In Sarath Chandar, Razvan Pascanu, and Doina Precup (eds.), *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pp. 243–254. PMLR, 22–24 Aug 2022. URL https://proceedings.mlr.press/v199/liu22a.html.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 17998–18031. Curran Associates, Inc., 2025a. doi: 10.52202/085713-0608. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1a22b912945fb7c0bdd079e792b31b6f-Paper-Conference.pdf.

Runze Liu, Jiashun Liu, Xu Wan, Yuqian Fu, and Ling Pan. When rl fails after sft: Rejuvenating model plasticity for robust sft-to-rl handoff. *arXiv preprint arXiv:2606.09932*, 2026.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, 7(2):181–194, 2025b.

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. *arXiv preprint arXiv:2308.08747*, 2023.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Martin Marek, Dongkyu Cho, Shikai Qiu, Rumi Chunara, Pavel Izmailov, and Andrew Gordon Wilson. Forgetting in language models: Capacity, optimization, and self-generated replay. *arXiv preprint arXiv:2605.26097*, 2026.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *Advances in Neural Information Processing Systems*, 38:132119–132138, 2026.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5716–5731, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.310. URL https://aclanthology.org/2024.acl-long.310/.

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto. Deep reinforcement learning with plasticity injection. *Advances in Neural Information Processing Systems*, 36:37142–37159, 2023.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

OpenAI. On the Navier–Stokes Millennium Prize problem. https://openai.com/index/navier-stokes-solution/, September 2026. Accessed 2026-09-20.

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.

Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. Task-specific skill localization in fine-tuned language models. In *International Conference on Machine Learning*, pp. 27011–27033. PMLR, 2023.

Yuxiao Qu, Amrith Setlur, Jasper Dekoninck, Edward Beeching, Jia Li, Ian Wu, Lewis Tunstall, Aviral Kumar, et al. Qed-nano: Teaching a tiny model to prove hard theorems. *arXiv preprint arXiv:2604.04898*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=GhVS8_yPeEa.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1810.11910.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in neural information processing systems*, 33:20378–20389, 2020.

Mutsumi Sasaki, Kouta Nakayama, Yusuke Miyao, Yohei Oseki, and Masaru Isonuma. Exclusive unlearning. *arXiv preprint arXiv:2604.06154*, 2026.

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

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, volume 2026, pp. 59839–59864, 2026.

Ilia Shumailov, Jamie Hayes, Eleni Triantafillou, Guillermo Ortiz-Jimenez, Nicolas Papernot, Matthew Jagielski, Itay Yona, Heidi Howard, and Eugene Bagdasaryan. Ununlearning: Unlearning is not sufficient for content regulation in advanced generative ai. *arXiv preprint arXiv:2407.00106*, 2024.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 32145–32168. PMLR, 2023.

Ali Taheri, Alireza Taban, Qizhou Wang, Shanshan Ye, Abdolreza Mirzaei, Tongliang Liu, and Bo Han. Forgetting: A new mechanism towards better large language model fine-tuning. *arXiv preprint arXiv:2508.04329*, 2025.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. Commonsenseqa 2.0: Exposing the limits of ai through gamification. *arXiv preprint arXiv:2201.05320*, 2022.

Sijun Tan, Michael Luo, Justin Wong, Colin Cai, Xiaoxiang Shi, William Yuan Tang, Manan Roongta, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Effective RL scaling of reasoning models via iterative context lengthening, 2026. URL https://openreview.net/forum?id=I6GzDCne7U.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Krithik Vishwanath, Jaden Stryker, Anton Alyakin, Daniel Alexander Alber, and Eric Karl Oermann. Medmobile: a mobile-sized language model with clinical capabilities. *BMJ Digital Health & AI*, 1, 2024. URL https://api.semanticscholar.org/CorpusID:282991758.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2026.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. In *International Conference on Learning Representations*, volume 2024, pp. 5385–5409, 2024.

Xiaoyu Xu, Xiang Yue, Yang Liu, Qingqing Ye, Huadi Zheng, Peizhao Hu, Minxin Du, and Haibo Hu. Unlearning isn't deletion: Investigating reversibility of machine unlearning in LLMs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=E5SVowO13b.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, volume 2025, pp. 29687–29707, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Kaiyan Zhang, Sihang Zeng, Ermo Hua, Ning Ding, Zhang-Ren Chen, Zhiyuan Ma, Haoxin Li, Ganqu Cui, Biqing Qi, Xuekai Zhu, et al. Ultramedical: Building specialized generalists in biomedicine. *Advances in Neural Information Processing Systems*, 37:26045–26081, 2024a.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024b.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Catastrophic failure of llm unlearning via quantization. In *International Conference on Learning Representations*, volume 2025, pp. 74925–74948, 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.

Hattie Zhou, Ankit Vani, Hugo Larochelle, and Aaron Courville. Fortuitous forgetting in connectionist networks. *arXiv preprint arXiv:2202.00155*, 2022.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Jinwon Lee, David Z. Pan, Zhangyang Wang, Yuandong Tian, and Kai Sheng Tai. The path not taken: RLVR provably learns off the principals. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025. URL https://openreview.net/forum?id=N75EWQQnb3.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. Can large language models transform computational social science? *Computational Linguistics*, 50(1):237–291, March 2024. doi: 10.1162/coli_a_00502. URL https://aclanthology.org/2024.cl-1.8/.

A TASKS AND DATASETS

Table 3 lists every task that appears in the paper, its role, how it is scored, and where the data come from. Three formats are used throughout. Generative tasks are scored by exact match of the extracted final answer against the reference (numbers for the math tasks, a letter for multiple-choice questions posed generatively, a normalized string for open-domain questions). Likelihood tasks are scored by choosing the answer option with the highest log-likelihood under the model. Open-ended tasks with no gradable answer (Dolly, Alpaca, ELI5) are tracked by the held-out language-modelling loss on their reference responses. Unless a table says otherwise, every task is capped at 500 items drawn with a fixed seed, so every checkpoint of a run and the base model score the same items.

Task (name in the paper)	Use	Format / metric	Source
Countdown	target	generative, exact match	synthetic, Appendix A.1
Countdown-3-4	in-domain transfer	generative, exact match	`Jiayi-Pan/Countdown-Tasks-3to4` (Pan et al., 2025)
DeepScaleR	target	generative, exact match	`agentica-org/DeepScaleR-Preview-Dataset` (Tan et al., 2026)
DAPO-17k	target (appendix)	generative, exact match	`BytedTsinghua-SIA/DAPO-Math-17k` (Yu et al., 2026)
MATH-500	math transfer	generative, exact match	Hendrycks et al. (2021)
GSM8K	math transfer / probe	generative, exact match	`openai/gsm8k` (Cobbe et al., 2021)
NuminaMath	math transfer / donor	generative, exact match	`AI-MO/NuminaMath-CoT` (LI et al., 2024)
FineProofs-SFT	SFT target	held-out loss; MATH-500 as proxy	`lm-provers/FineProofs-SFT` (Qu et al., 2026)
OpenBookQA	target / probe	generative letter (target) or likelihood (probe)	`allenai/openbookqa` (Mihaylov et al., 2018)
AI2-RC (ARC-Challenge)	target / transfer / donor	generative letter or likelihood	`allenai/ai2_arc` (Clark et al., 2018)
CommonsenseQA	transfer / donor	likelihood	Talmor et al. (2019)
CSQA2	donor	generative, exact match	`tasksource/commonsense_qa_2.0` (Talmor et al., 2022)
QASC	donor	generative, exact match	`allenai/qasc` (Khot et al., 2020)
BoolQ	donor	generative, exact match	`google/boolq` (Clark et al., 2019)
StrategyQA	target / donor	generative yes/no	`ChilleD/StrategyQA` (Geva et al., 2021a)
HellaSwag	probe / donor	likelihood	`Rowan/hellaswag` (Zellers et al., 2019)
MMLU (STEM, humanities, social sciences, other)	probe / donor	likelihood	`cais/mmlu` (Hendrycks et al., 2020)
TriviaQA	donor	generative, exact match	`mandarjoshi/trivia_qa` (no context) (Joshi et al., 2017)
HotpotQA	donor	generative, exact match	`hotpotqa/hotpot_qa` (Yang et al., 2018)
ECQA	donor	generative, exact match	`yangdong/ecqa` (Aggarwal et al., 2021)
Dolly	donor	held-out loss	`databricks/databricks-dolly-15k` (Conover et al., 2023)
Alpaca	donor	held-out loss	`tatsu-lab/alpaca` (Taori et al., 2023)
ELI5	donor	held-out loss	`sentence-transformers/eli5` (Fan et al., 2019)
XSum	donor	generative, ROUGE-L	`EdinburghNLP/xsum` (Narayan et al., 2018)
CaseLawQA (Lawma)	SFT target (appendix) / probe	generative letter	`ricdomolm/lawma-instructions` (Dominguez-Olmedo et al., 2025)
Countdown-H	target variant (Appendix B.1)	generative, exact match	synthetic, Appendix A.1 (six numbers, four or more operations)
ARC-Easy	probe	likelihood	`allenai/ai2_arc` (ARC-Easy) (Clark et al., 2018)
MedQA	probe	generative letter	`GBaker/MedQA-USMLE-4-options`
MedMCQA	probe	generative letter	`openlifescienceai/medmcqa`
ReClor	probe	generative letter	`metaeval/reclor`
ProofWriter	probe	generative, exact match	`tasksource/proofwriter`
DROP	probe	generative, exact match	`ucinlp/drop`
Text corpora (FineWeb 2025, arXiv 2026, PubMed and arXiv from the Pile)	probe	held-out loss	held-out documents of each corpus

Table 3: Every task used in the paper. "Donor" marks tasks that serve as forgetting tasks for LRA or as address sources for ANTIGEN; "probe" marks tasks evaluated only to measure forgetting, including the tasks scored only in the 38-task evaluation of Appendix B.1.

A.1 THE COUNTDOWN TASK

Each Countdown instance deals four integers and a target between 100 and 999. The model must combine some or all of the dealt numbers with $+$, $-$, $\times$ and integer division, using each number at most once, to reach the target. Instances are generated by planting a solution that uses three or four of the numbers, so every instance is solvable, and a verifier grades the final expression by evaluating it and comparing with the target. The training set holds 10,000 seeded instances and the evaluation split 500 further instances that never appear in training. The reward is the exact-match verdict of the verifier. Prompts use the system instruction "Solve the following puzzle. Reason step by step, and put your final answer within `\boxed{}`" and the model runs in non-thinking mode with a 1,024-token completion budget, since the thinking mode of the 0.8B model does not terminate within any practical budget. Countdown-3-4 is the public Countdown-Tasks-3to4 set of Pan et al. (2025), in which every instance has three or four dealt numbers. We hold out 500 of its instances as a zero-shot in-domain transfer set. Two harder variants exist in our code: one requires every dealt number to be used, and one deals six numbers and keeps only targets whose shortest solution needs at least four operations. The curriculum baselines of Appendix B train on them, and the second is scored as Countdown-H among the target variants of Appendix B.1.

A.2 EVALUATION DETAILS FOR TABLE 1

All pairs in Table 1 are scored under one protocol (500-item caps; multiple-choice probes scored by likelihood). Two caveats apply to MATH-500. The generation budget differs within some pairs (base / specialist): Qwen2.5-Math-1.5B (8k / 2k tokens); Qwen2.5-Math-7B (8k / 2k tokens); DeepSeek-R1-Distill-Qwen-1.5B (2k / 32k tokens); DeepSeek-R1-Distill-Qwen-7B (2k / 32k tokens); Lawma-8B (4k / 8k tokens). For Pythagoras-Prover-4B and Qwen3-4B-SFT-science, at least 99% of MATH-500 completions reach the 32k-token budget without terminating, so those scores are lower bounds.

Table 4 gives the exact model identifiers behind Table 1.

Domain	Base model	Specialist	Training
math	`Qwen/Qwen3-4B-Thinking-2507`	`lm-provers/QED-Nano`	SFT+RL
math	`Qwen/Qwen3-4B`	`POLARIS-Project/Polaris-4B-Preview`	RL
math	`Qwen/Qwen3-4B`	`Pythagoras-LM/Pythagoras-Prover-4B`	SFT+RL
math	`Qwen/Qwen3-4B`	`NotoriousH2/Qwen3-4B-Countdown-RLVR`	RLVR
math	`Qwen/Qwen2.5-1.5B`	`Qwen/Qwen2.5-Math-1.5B`	SFT
math	`Qwen/Qwen2.5-7B`	`Qwen/Qwen2.5-Math-7B`	SFT
reasoning	`Qwen/Qwen2.5-Math-1.5B`	`deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B`	distillation
reasoning	`Qwen/Qwen2.5-Math-7B`	`deepseek-ai/DeepSeek-R1-Distill-Qwen-7B`	distillation
code	`Qwen/Qwen2.5-1.5B`	`Qwen/Qwen2.5-Coder-1.5B`	SFT
code	`Qwen/Qwen2.5-7B`	`Qwen/Qwen2.5-Coder-7B`	SFT
science	`Qwen/Qwen3-4B`	`graf/Qwen3-4B-SFT-science-1e-5`	SFT
agentic search	`Qwen/Qwen3-4B-Thinking-2507`	`janhq/Jan-v1-4B`	RL
medical	`Qwen/Qwen3-4B-Thinking-2507`	`qvac/MedPsy-4B`	SFT
medical	`NousResearch/Meta-Llama-3-8B`	`TsinghuaC3I/Llama-3-8B-UltraMedical`	SFT
legal	`NousResearch/Meta-Llama-3-8B`	`ricdomolm/lawma-8b`	SFT
multilingual	`Qwen/Qwen3-4B-Base`	`McGill-NLP/AfriqueQwen-4B`	SFT

Table 4: Hugging Face identifiers of the base and specialist models in Table 1.

B ANTIGEN: ALGORITHM, HYPERPARAMETERS AND ABLATIONS

Settings. On Countdown the addresses are the rank-16 LRA projectors of the 21-donor broad suite, stacked to rank 336, attached to every linear projection in layers 12–23, with plasticity scale $S = 25$. The same recipe is used from the base model on DeepScaleR at 0.8B and 2B. On OpenBookQA the addresses come from the five-task knowledge suite (rank 80) with $S = 12.5$, that suite having been selected by its Fisher ratios and LRA outcome on AI2-RC. In the reported arms the adapters are not merged before training (do_merge `false`), and $\hat{B}$ is folded at every saved checkpoint, so every evaluated checkpoint is a plain model of the original architecture. On Countdown $\hat{B}$ spans the 93 linear projections of layers 12–23 at rank 336: 67.2M trainable parameters, 8.9% of the model's 752M weights, added during training only.

Algorithm 1 Anti-General Intelligence (ANTIGEN)

Require: Specialist model W; target task $\mathcal{T}$; donor tasks $\mathcal{T}'_1, \ldots, \mathcal{T}'_T$; per-task rank k; do_merge (`true`/`false`); plasticity scale S; fold interval F

for $i = 1, \ldots, T$ **do** ▷ address derivation (one-time)
  train adapter (A_i, B_i) on $\mathcal{T}'_i$ by LRA (equation 1); keep A_i, discard B_i
  if do_merge **then** $W \leftarrow W + B_i A_i$ ▷ optionally, induce forgetting
  end if
end for
$A \leftarrow [A_1; \cdots; A_T]$ ▷ frozen address matrix, rank Tk
$\hat{B} \leftarrow 0$, trainable, with layer output $h = Wx + s\hat{B}Ax$ where $s \propto S$
for step $t = 1, 2, \ldots$ **do**
  update $[W \mid \hat{B}]$ on $\mathcal{T}$; $\hat{B}$ learns $\sim S\times$ faster through s ▷ specialization via e.g., GRPO
  if $t \bmod F = 0$ **then** $W \leftarrow W + s\hat{B}A$; $\hat{B} \leftarrow 0$ ▷ fold-and-reset keeps rollouts on-policy
  end if
end for
return W ▷ final fold keeps model architecture

Hyperparameter search. Table 6 lists every arm run from the Countdown specialist. The ANTIGEN rows cover the plasticity scale ($10\times$, $25\times$, $40\times$, and $25\times$ ramped in over 200 steps), the donor family (broad 21, knowledge 6, non-math reasoning 2), merging the ascent adapters before training, and addresses extracted by descent instead of ascent. A $10\times$ scale gives no gain over GRPO; a ramped $25\times$ single run is competitive with the constant $25\times$ arm, but only the constant arm was replicated across seeds, so it is the configuration reported everywhere else; and merging before training costs 3–10 points on every suite we tried.

B.1 ACCURACY AND LOG-PROBABILITIES

Figures 11–14 score the base model, each specialist and the end checkpoint of each arm on 38 tasks of 500 items: the target and its in-domain variants, the ANTIGEN arm's donors that the suite covers (19 of the 21 Countdown donors; MMLU-Aux and XSum are not scored) and probes that were never donors. The top rows show the change in graded accuracy; the bottom rows the change in a log-probability view of the same items — the gold option's log-likelihood for likelihood-scored tasks, the mean per-token log-probability of the reference solution for generative tasks (of the model's own completion where no reference exists, hatched), the gold letter's log-probability at the answer position for MedQA, MedMCQA and CaseLawQA, and the negated held-out loss for the text corpora. Group means are printed above each panel; green marks accuracy gains that come from more completions finishing inside the token budget (below).

Continued GRPO leaves the rest of the model where the specialist left it. From θ_S to step 2400 the donor and probe accuracies move by $+0.1$ and -0.1 points, their log-probabilities by -0.007 and -0.009 nats, and no task by more than 0.07 nats. ANTIGEN *lowers the likelihood of non-target text without lowering its accuracy.* Over the same window its donor and probe log-probabilities fall by 0.115 and 0.112 nats while their accuracies rise by 1.2 and 2.2 points. The fall is focal: it lands on the likelihood-scored donors whose options are full sentences (HellaSwag -0.73, OpenBookQA -0.46, AI2-RC -0.32, ARC-Easy -0.21 nats), while the single-token MMLU options and the text corpora move by at most 0.04 nats and the letter-choice tasks move by less than 0.03 nats or rise (MedQA $+0.27$, CaseLawQA $+0.10$); and it removes mass from the gold option without blurring the choice, since the gold option's share among the options rises (MMLU-STEM $0.37 \rightarrow 0.39$, CommonsenseQA $0.40 \rightarrow 0.46$) and multiple-choice accuracy is flat. Plain LoRA and random addresses in the same container shift log-probabilities too (-0.085/-0.128 and -0.062/-0.079 nats on donors/probes; Figure 12), so most of the shift comes with the high-plasticity adapter itself; on the sentence-option donors, however, ANTIGEN's drop is two to four times LoRA's (-0.18, -0.16 and -0.14 nats), matching the donor-specific Fisher compression of Appendix C.3. *The accuracy gains on the reasoning probes are gains in termination, not in hit rate.* On BoolQ, ReClor, ProofWriter and DROP, ANTIGEN gains 16.0, 11.6, 9.0 and 5.1 points over θ_S while the share of completions finishing inside the budget rises from 53, 16, 24 and 30% to 76, 41, 38 and 47%; accuracy among the finished completions is flat or lower, so the finished share explains 90–140% of each gain, and chains shorten by 14–24% on the short-answer tasks (BoolQ 213 to 181 tokens). Continued GRPO

changes no finished share, LoRA shows the effect at a smaller size and random addresses barely; we therefore do not read these tasks as transfer. The from-base settings repeat the pattern at a larger scale: on DeepScaleR at step 1200, ANTIGEN leads GRPO by 3.4, 2.2 and 4.1 points on the target, donor and probe groups while its log-probabilities fall by 0.62, 0.22 and 0.16 nats against GRPO's -0.07, $+0.01$ and -0.01, and the reference solutions of the math targets lose 0.3–1.1 nats per token as their accuracy rises, so the model's solutions diverge from the reference style rather than from the answers; on OpenBookQA at step 3000 the donors fall by 0.14 nats against -0.002 under GRPO and the probes do not move.

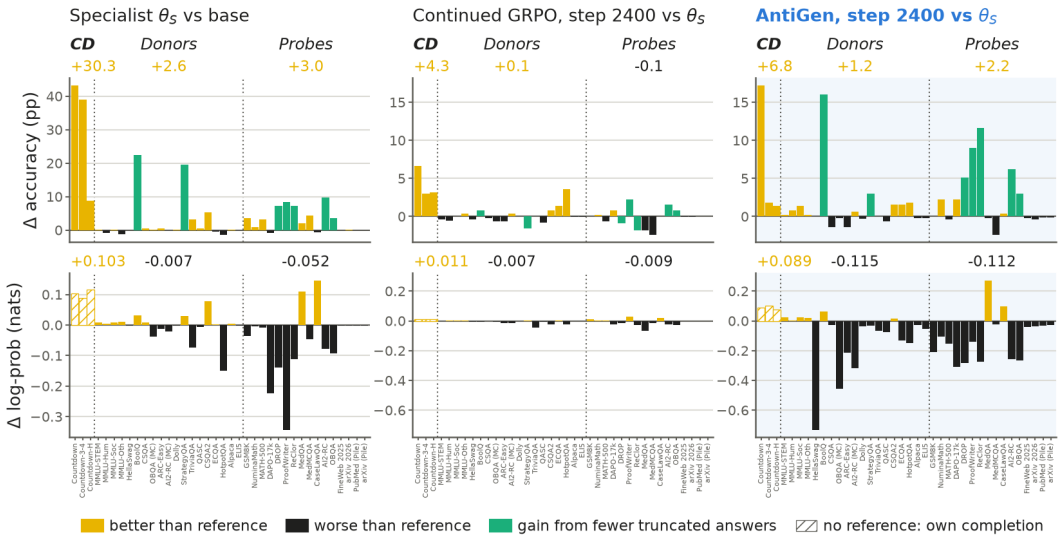


Figure 11: **Countdown:** change in accuracy (top) and in the log-probability view (bottom) per task, for the specialist against the base model (left) and for continued GRPO and ANTIGEN at step 2400 against θ_S (middle and right, shared axes). Gold: better than the reference; black: worse; green: an accuracy gain that comes from fewer truncated answers; hatched: no reference solution, so the model's own completion is scored.

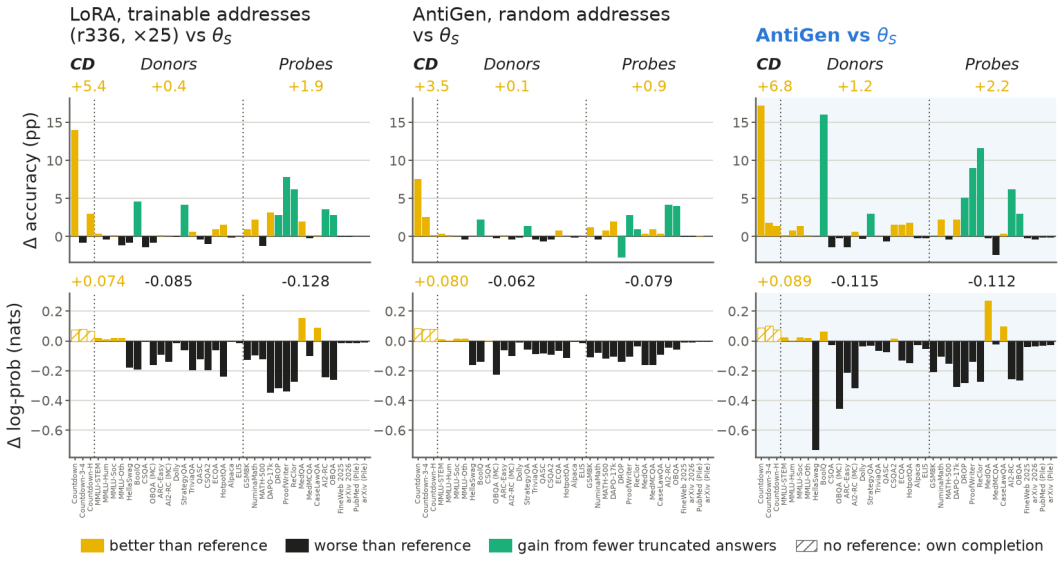


Figure 12: **Countdown, container ablations:** as Figure 11, for plain LoRA (trainable A), random addresses and ANTIGEN at step 2400, all against θ_S on shared axes.

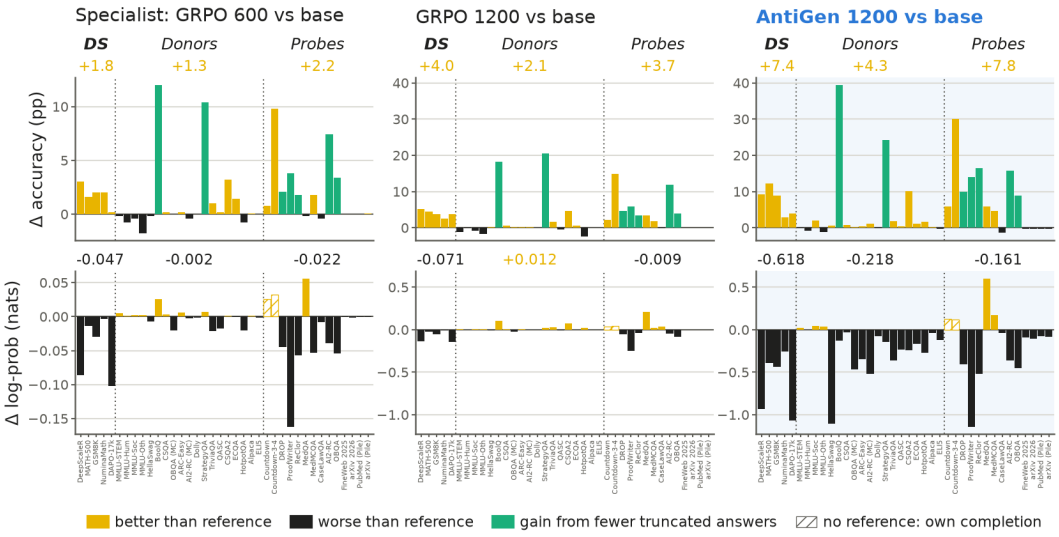


Figure 13: **DeepScaleR (0.8B, from base, seed 43):** as Figure 11, for the specialist (GRPO step 600), continued GRPO (step 1200) and ANTIGEN (step 1200), all against the base model.

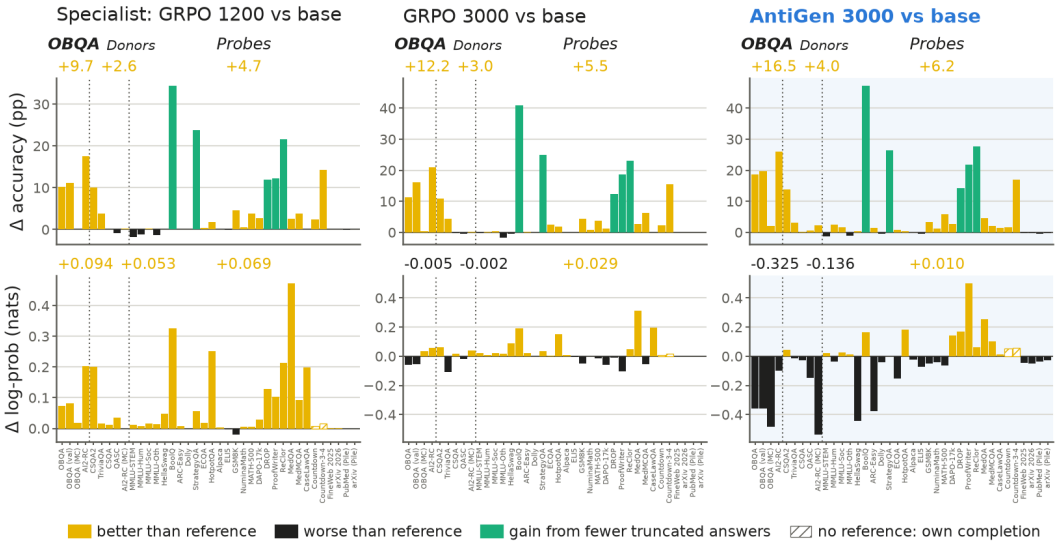


Figure 14: **OpenBookQA (0.8B, from base, seed 43):** as Figure 11, for the specialist (GRPO step 1200), continued GRPO (step 3000) and ANTIGEN (step 3000), all against the base model.

B.2 ADAPTER ABLATIONS

Random addresses replace the ascent-derived A by a fixed random matrix of the same rank and initialization scale, with the same output scale. *Plain LoRA* additionally makes A trainable at the learning rate of the content matrix. *Full rank* removes the low-rank structure and trains a zero-initialized full matrix per projection. *Layer-wise learning rate* raises the learning rate of layers 12–23 by $10\times$ or $25\times$ with no adapter at all. Only the low-rank adapters help: the full-rank adapter loses 2.3 points and the layer-wise rate 4.1. All three low-rank arms beat the control over three seeds each at 2400 steps, by $+8.2$ for the ascent-derived addresses, $+6.6$ for plain LoRA and $+5.3$ for random addresses on the mean of the per-seed peaks, and the ascent-derived arm is both the highest and the tightest: $62.4 \pm 1.1\%$ against 60.8 ± 2.8 and 59.5 ± 2.2, with the control at 54.2 ± 1.4 (mean and standard deviation of the three per-seed peaks). Their window means are within 1.5 points of each other, so the separation is in the height of the best checkpoint rather than in the level of the whole window.

How far the ascent moves the addresses. The ceiling that stops the ascent after 10–20 steps leaves A close to its random initialization: the relative drift $\|A - A_0\|_F / \|A_0\|_F$ is 0.055 on Countdown, 0.057 on DeepScaleR and 0.047 on OpenBookQA per donor block, the update has effective rank 3.4 of a possible 16, and two seeds agree on it at cosine 0.93 (Figure 15). The reported ANTIGEN and random-address arms therefore differ mainly in which subspace they occupy, consistent with their 2.9-point separation. Driving the ascent further (no ceiling, learning rate 10^{-3}, 100 steps; drift 0.63, effective rank 5.2) does not help: over three seeds the driven arm matches ANTIGEN to step 1800 ($+2.1$ peak over the control) and then falls below the control (-8.9 mean over steps 1620–2400; Table 6), so the small rotation is not a shortfall of the recipe, and a large one destabilizes training.

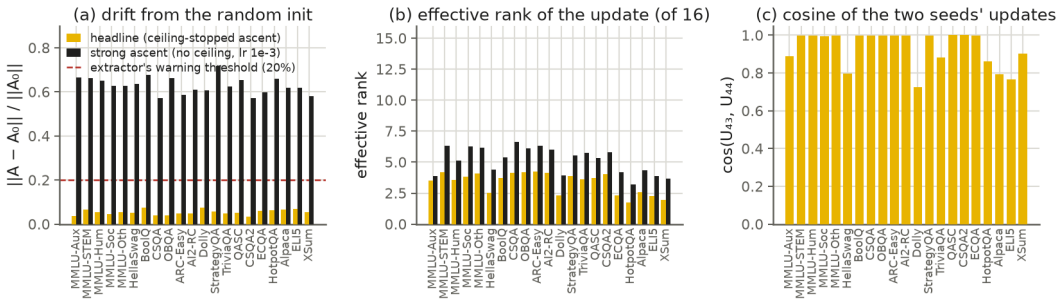


Figure 15: Per-donor anatomy of the Countdown address matrix. (a) Relative drift of each donor's block from its random initialization, for the ceiling-stopped ascent used throughout the paper and for a driven ascent; the dashed line is the 20% threshold below which the extractor warns that the addresses approximate the random ablation. (b) Effective rank of the update, of a possible 16. (c) Cosine between the updates of two seeds.

Attaching to an existing specialist. ANTIGEN does not have to be run from the base model. Attaching the adapter to the saturated control instead reaches the same peak: on DeepScaleR, 34.8 against 34.0 for the from-base arm and 32.1 for continued GRPO over steps 600–1200; on OpenBookQA, 73.8 against 73.7 and 70.3 over steps 1200–3000. Window means are 1.0 to 1.6 points below the from-base arms and 2.1 points above the controls. These are single seeds and are drawn as the third curve in Figure 16.

B.3 RL PLASTICITY BASELINES

Each baseline is a true resume of θ_S under the GRPO recipe of Appendix F with one change. *ProRL*-style resets (Liu et al., 2025a) add a KL penalty to a reference policy and hard-reset the reference and the optimizer either every 100 steps or on the same reward plateau that triggers LRA merges. *Clip-Cov* (Cui et al., 2025) removes the gradient of the 2×10^{-4} fraction of completion tokens with the highest covariance between log-probability and advantage. *High-entropy tokens* (Wang et al., 2026) applies the policy loss only to the 20% highest-entropy tokens of each batch. *KL replay* draws

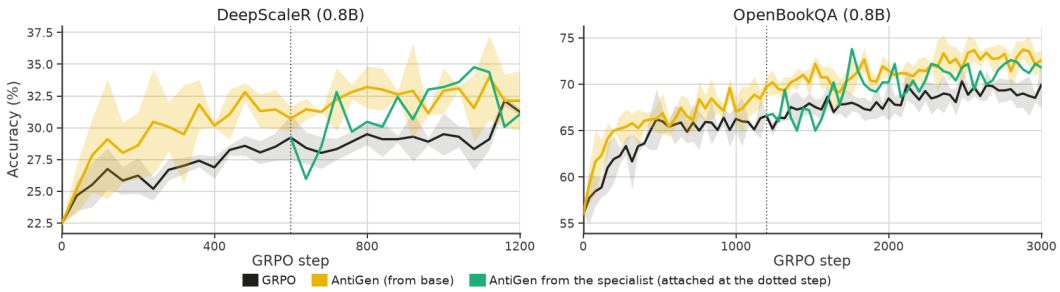


Figure 16: Target accuracy on DeepScaleR (0.8B) and OpenBookQA for GRPO and ANTIGEN from the base model (three seeds, bands of one s.d.) and for ANTIGEN attached to the saturated control at the dotted step (one seed).

1,024 samples per rank from the frozen specialist at the first step and adds the token-level KL to the specialist on three replayed sequences to every loss. *ReDo* (Sokar et al., 2023) re-initializes the 5% most non-math-selective MLP units of layers 12–23 before training. Two *data curricula* keep plain GRPO but change the instances. Over the full 1620–2400 window the best of these is ProRL at $+2.2$ peak and $+1.0$ mean; Clip-Cov and high-entropy tokens are within noise of the control, and ReDo collapses the target.

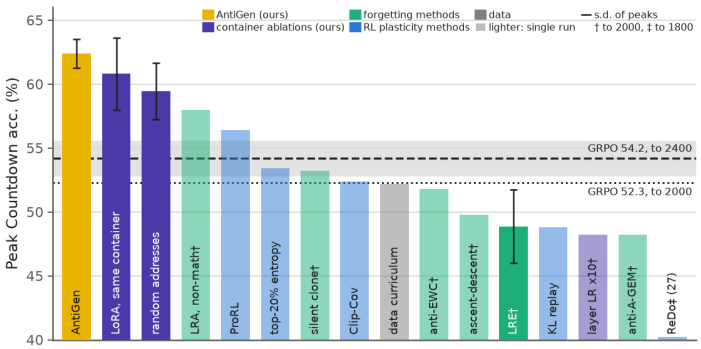


Figure 17: Peak Countdown accuracy of every method run from θ_S over the longest window its runs reach: steps 1620–2400, or 1620–2000 (†) and 1620–1800 (‡) for arms not extended. Each run contributes its best scored checkpoint in the window; bars average these peaks across runs and error bars give their standard deviation over runs. Lighter bars are single runs. The dashed and dotted lines are the 3-seed GRPO control's mean peak over the two windows.

C CHARACTERIZATION DETAILS

The characterizations of Section 5.1 are computed at the saturated Countdown specialist θ_S (1600 GRPO steps) from eight batches of eight 1,024-token sequences per task, drawn from each task's training split with a fixed seed. Countdown carries no reference solutions, so its texts are a fixed set of verified-correct completions: the 49 correct answers among 100 samples of the specialist, each prepended to its prompt. The same set is scored, teacher-forced, at every checkpoint and for every arm, so the Countdown gradients and Fisher are properties of the loss on fixed texts and cannot track a policy's own confidence.

C.1 ROLLOUT ENTROPY

The entropies quoted in the main text are per-token means over the training rollouts, logged at every optimizer step at the sampling temperature of 0.8 and drawn as a trailing 20-step mean. ANTIGEN does not add exploration. Its rollouts are less entropic than the control's throughout the window, 0.113 ± 0.016 against 0.172 ± 0.004 nats per token over steps 1620 to 2400 (three seeds each), with

the specialist at 0.193, and plain LoRA and random addresses in the same adapter fall by the same amount (0.124 and 0.110). The two baselines that raise entropy, Clip-Cov (0.202) and high-entropy tokens (0.185), gain nothing (Table 7). The gain therefore coincides with a sharper policy rather than a more exploratory one.

C.2 Activation maps

For every MLP we record the mean absolute activation of each intermediate unit over the task's tokens, giving a tasks-by-units matrix per layer; the mean-subtracted maps subtract each unit's average over tasks. Units are sorted by the task for which their differential activation is largest, so task-selective units form contiguous blocks. Selectivity is present at every depth and does not concentrate late: the shared unit means explain 0.44 to 0.94 of the variance, the median across-task coefficient of variation is 0.30 to 0.35 in every layer, and 3 to 7% of units give more than a quarter of their activation to one task (a uniform unit gives one tenth). The differential structure is also the same size relative to each layer's mean activation at every depth (0.26 to 0.30). Late layers look stronger in the raw maps only because their activations are four times larger in absolute terms.

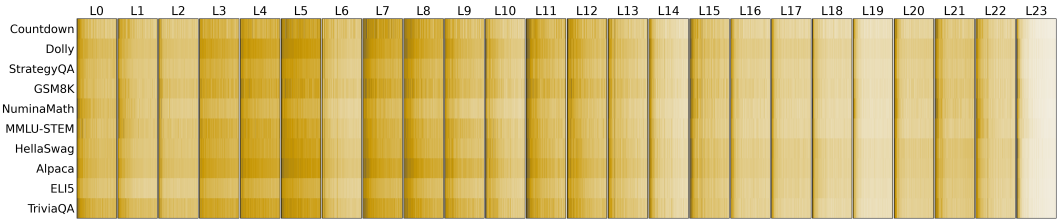


Figure 18: Raw MLP activation maps of θ_S for the layers of Figure 8. Rows are tasks and columns are units, sorted as in the main text.

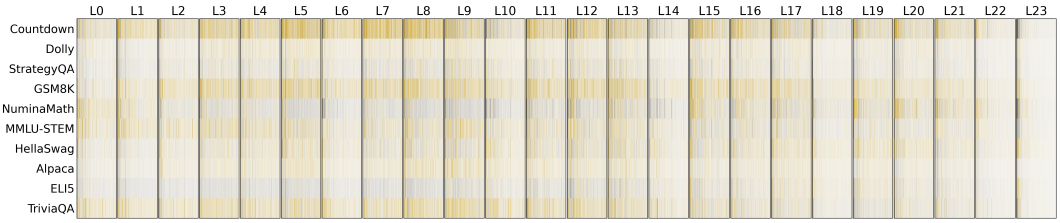


Figure 19: Mean-subtracted MLP activation maps of θ_S for the layers of Figure 8. Subtracting the across-task unit mean removes the shared high-traffic units and exposes the selective structure.

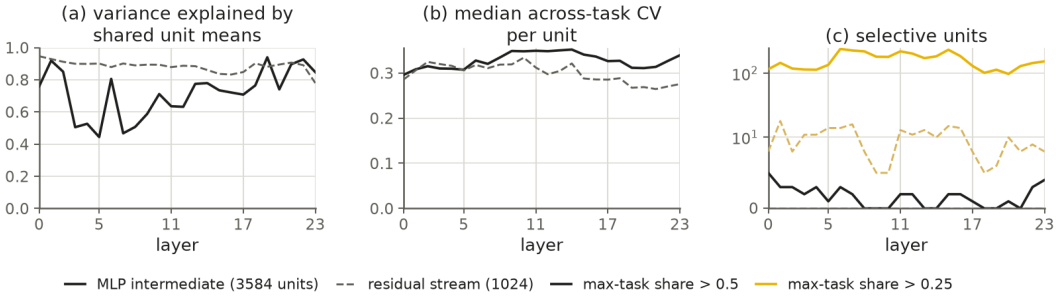


Figure 20: Task selectivity by layer at θ_S (solid, the 3,584 MLP units per layer; dashed, the 1,024 residual-stream dimensions). (a) Variance explained by the shared across-task unit means. (b) Median across-task coefficient of variation per unit. (c) Units whose largest task share exceeds 0.5 (black) or 0.25 (gold). Selectivity is present at every depth with no concentration in the late layers.

C.3 FISHER INFORMATION AT THE SPECIALIST AND AFTER TRAINING

Fisher information. We use the empirical diagonal Fisher, the squared gradient of the task loss averaged over the same batches, summarized per layer by the mean over that layer's parameters. Ratios divide a task's per-layer Fisher by the target's, and "late-layer" summaries average the ratio over the last six layers. Fisher magnitudes depend on each task's loss format and on how surprising its data are under θ_S, so we read them as where the loss is sensitive rather than as a parameter count. Across settings, the Fisher of non-target tasks concentrates in the late layers, exceeding the target's by one to two orders of magnitude at the specialist (10–50$\times$ on Countdown, up to 170$\times$ for TriviaQA on DeepScaleR), while tasks that share structure with the target sit at or below parity.

Figure 21 shows the per-layer Fisher of every characterized task at θ_S and its ratio to Countdown's.

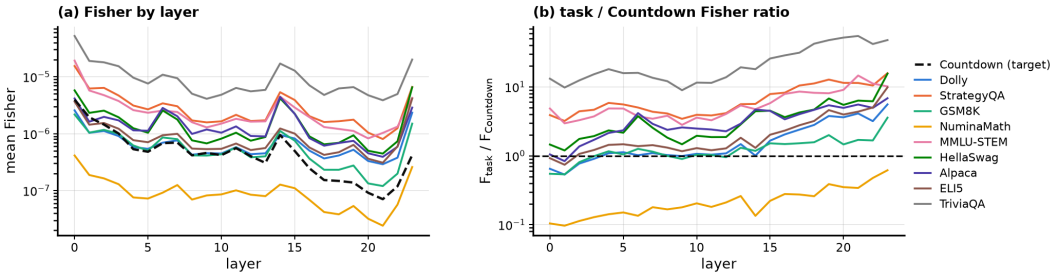


Figure 21: Per-layer Fisher of θ_S (left, Countdown dashed) and the task-to-Countdown Fisher ratio (right). Non-math probes hold 10–50$\times$ the target's late-layer Fisher, while the math tasks sit below $1\times$.

Before and after training. Figure 22 tracks the same ratios through each transition, as geometric means over the seven non-math tasks and the two math tasks, and Figure 23 shows the nine tasks separately. On Countdown the base-to-specialist step raises the early-layer ratios and leaves the late layers unchanged, continued GRPO changes nothing, and ANTIGEN lowers the late-layer ratios of the donors by 16% (geometric mean over the seven characterized donors) while the two math tasks, which are not donors, move by 6%. DeepScaleR shows the same pattern with a target that is itself mathematical: continued GRPO leaves every ratio in place, ANTIGEN lowers the donor ratios by 5% and raises GSM8K and NuminaMath by 78%, so the late-layer curvature shifts toward the tasks the target shares structure with. On OpenBookQA the five donors fall by 8% while the eight non-donor probes move by -2%, and the seed-44 arm shows the same separation from a higher baseline. The DeepScaleR and OpenBookQA ANTIGEN checkpoints are the from-base runs at steps 1200 and 2400, compared with each control at its specialist point; for DeepScaleR, comparing with the base model instead gives the same numbers, since the base-to-specialist step moves the late-layer ratios by only 2%.

Is the compression specific to the addresses? Characterizing the endpoints of the adapter ablations under the same protocol shows that it is not. Relative to the specialist, the late-layer donor ratios fall by 16% under ANTIGEN at step 2000, by 8% under plain LoRA at the same step, and by 27% and 18% under plain LoRA and random addresses at step 2400, while continued GRPO moves them by 0% (Figure 24). Every arm that trains a high-plasticity low-rank adapter in layers 12–23 compresses the ratios, and the arms that train longest compress most; the ascent-derived addresses are not required for the effect. The math tasks behave differently from the donors in every adapter arm, rising slightly while the donors fall, which is the asymmetry the recruitment reading rests on.

C.4 GRADIENT ALIGNMENT AND SUBNETWORK OVERLAP

Gradient alignment. For each task we average the loss gradient over the eight batches and report the cosine between the averaged Countdown gradient and the averaged gradient of every other task, over all transformer parameters. Three reference points set the scale. Two disjoint halves of the Countdown set agree at 0.978, so the measurement itself is stable. Two *related* probe tasks agree at 0.134 (GSM8K with NuminaMath) and 0.153 (StrategyQA with Dolly). Unrelated random di-

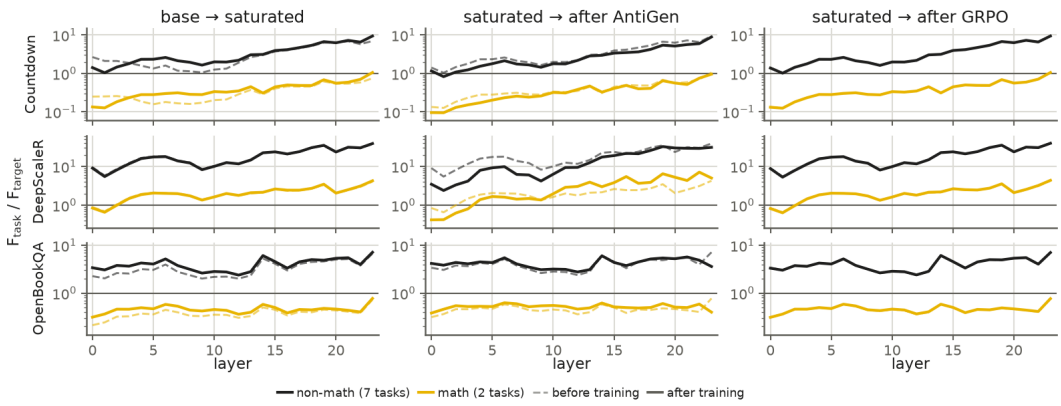


Figure 22: Task-to-target Fisher ratio per layer before (dashed) and after (solid) each transition, as geometric means over the seven non-math tasks (black) and the two math tasks (gold). Rows are settings; columns are the base-to-specialist, specialist-to-ANTIGEN and specialist-to-GRPO transitions. Non-target curvature sits one to two orders of magnitude above the target's in the late layers; ANTIGEN compresses it there and GRPO does not.

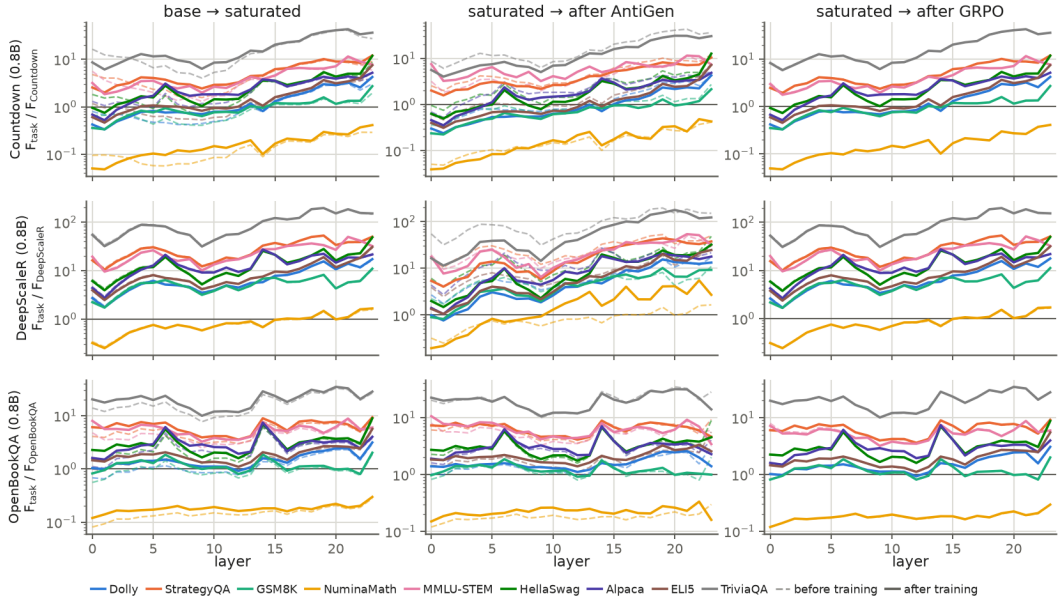


Figure 23: Per-task detail of Figure 22: the task-to-target Fisher ratio per layer before (dashed) and after (solid) each transition, for each of the nine characterized tasks.

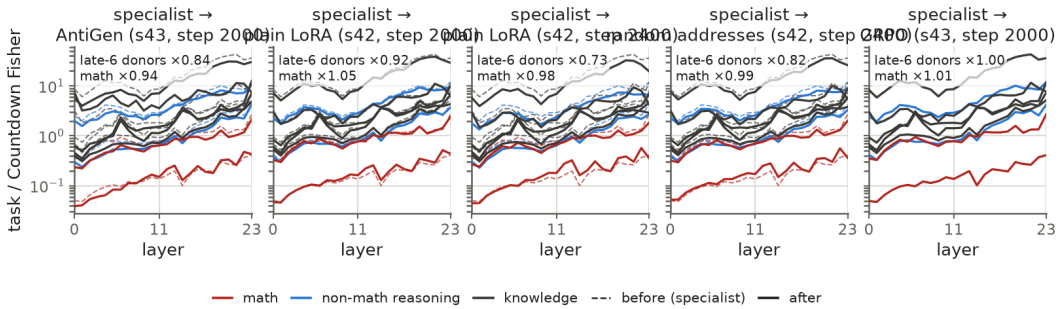


Figure 24: Task-to-Countdown Fisher ratio per layer before (dashed) and after (solid) each adapter arm, with the late-layer geometric means annotated. Every low-rank adapter compresses the donor ratios; continued GRPO does not.

rections would give $1/\sqrt{d} \approx 4 \times 10^{-5}$ in $d = 7.5 \times 10^8$ parameters. The cross-task cosines of at most 0.06 are therefore far above chance but well below both the task's own self-consistency and the agreement of two related tasks.

Shared parameters despite orthogonal gradients. A near-zero cosine means a Countdown step changes another task's loss only to second order; it does not mean the two tasks use different parameters. Taking the top 1% of parameters by diagonal Fisher for each task and intersecting with Countdown's set gives Jaccard overlaps of 0.12 to 0.19, against 0.005 for independent sets of the same size (Figure 26); at the top 0.1% the overlaps are 0.12 to 0.21 against 0.0005. The two mathematical tasks overlap most (NuminaMath 0.187, GSM8K 0.171) and MMLU-STEM least (0.120), the same ordering as the LRA interference of Section 5.

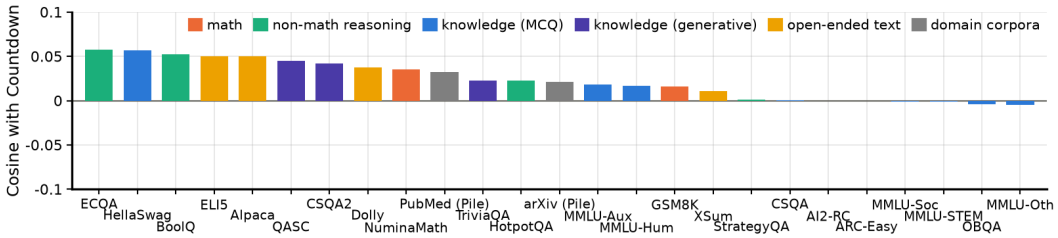


Figure 25: Gradient cosine between Countdown and each of the 25 candidate tasks at θ_S. Bars are sorted and coloured by task family; the largest value is 0.057, against 0.978 between two disjoint halves of the Countdown set.

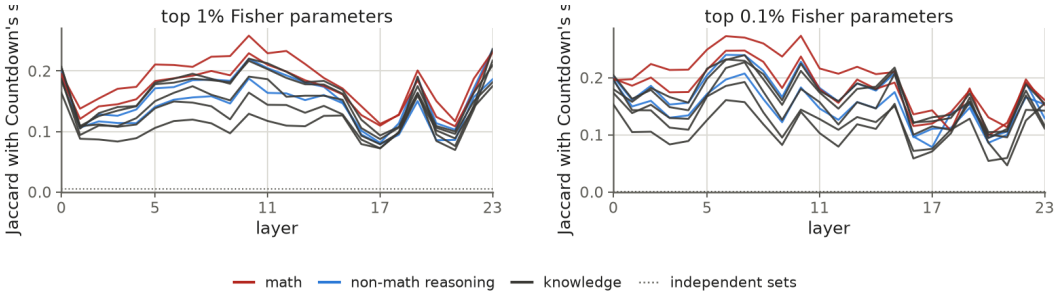


Figure 26: Jaccard overlap between each task's top-Fisher parameter set and Countdown's, per layer, at θ_S. Independent sets of the same size would give the dotted line. Every probe shares far more of the target's important parameters than chance at every depth, most of all the mathematical tasks.

C.5 Representational similarity

Linear centered kernel alignment (CKA) between the residual-stream activations of each task and those of Countdown, computed per layer at θ_S on the same batches, stays between 0.1 and 0.3 across the whole stack (Figure 27). The open-ended text tasks (ELI5, Dolly) are the most similar to Countdown and NuminaMath the least, and similarity grows slowly with depth. The representations the tasks induce therefore remain largely distinct even in the late layers where the non-target tasks hold most of their Fisher information, consistent with the task-selective activation blocks above.

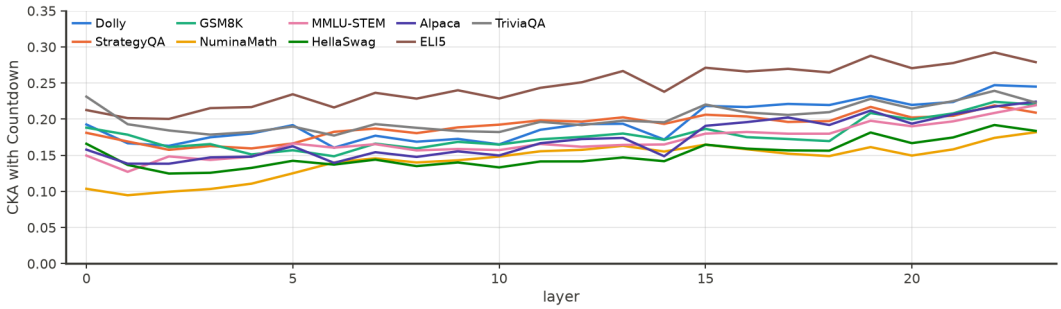


Figure 27: Linear CKA between each task's residual-stream activations and Countdown's, per layer, at θ_S.

D LOW-RANK ASCENT: DETAILS, DONOR SUITES AND ALTERNATIVES

Recipe. Every LRA adapter has rank 16 with LoRA scale $\alpha = 32$ and is attached to every linear projection of every transformer layer. For each donor the adapter is trained by gradient ascent on that task's loss with Adam at learning rate 10^{-4}, batches of eight 1,024-token sequences and at most 100 steps, stopping as soon as the task loss exceeds a ceiling of 8 nats per token; in practice the ceiling is reached within 10–20 steps. The adapter is then merged into the weights with merge weight $\beta = 0.3$ and discarded. Merging happens either on a fixed period or on a plateau schedule; under the latter, one merge is guaranteed at the start of the run and a further merge fires whenever the mean reward over the second half of a 20-step window is within 0.02 of the mean over the first half, with a cooldown of 40 steps. Each donor contributes at most 10,000 training examples.

Donor suites and windows. The suites behind Table 2 follow one recipe across targets. The math suite is GSM8K and NuminaMath. The non-math reasoning suite is Dolly and StrategyQA, with Countdown added when StrategyQA is the target. The knowledge suite is CSQA2, TriviaQA, OpenBookQA, CommonsenseQA and QASC for the from-base targets, ARC-Challenge replacing OpenBookQA when OpenBookQA is the target, and the MMLU subsets with HellaSwag and commonsense tasks for Countdown. The broad suite is the union of the 20–21 donors marked in Table 3. Each LRA arm and its control are compared over the same window of scored checkpoints: 1620–2000 for Countdown (from the 1600-step specialist), 1720–2000 for AI2-RC, 640–1000 for StrategyQA and DeepScaleR (600-step specialists), 1240–1400 for OpenBookQA (1200-step specialist), 150–300 SFT steps for Lawma, 40–300 for DAPO-17k from base, and 10–36 SFT steps for FineProofs.

Reading Table 2. Donor performance drops in every setting, so LRA induces the intended forgetting; the cost to the target depends on the donor family. Math donors reliably degrade the Countdown and DeepScaleR targets (8–11 points) and damage collateral tasks broadly (9–14 points), consistent with the shared curvature of Section 5.1. Knowledge and non-math reasoning donors leave the target within noise. The largest positive cell, Countdown with non-math donors, is a single plateau-scheduled run and does not replicate: a matched re-run of the same recipe loses 4.5 points and the two pool to -0.5 (Table 7), so we do not read the cell as a reproducible gain. The only other target gains above two points are FineProofs with knowledge donors ($+2.7$) and one periodic-scheduled Countdown run ($+5.7$ peak over steps 1620–2000, also a single run). Collateral damage under these donors is smaller but not absent (0.7–7.7 points). We therefore read Table 2 as evidence that focal forgetting is achievable, and that its effect on the target is neutral at best unless the donors share structure with the target, in which case it is harmful.

Alternative forgetting methods. Beyond LRA we implemented five further ways of inducing forgetting during specialization, all evaluated on Countdown from θ_S under the protocol of Appendix B; their outcomes are in Table 7. *Low-rank extraction* (LRE) trains the same rank-16 adapter by descent on the donor task and subtracts it from the weights. *Ascent–descent* adds a scaled ascent term on donor batches to every GRPO step instead of merging adapters. *Anti-EWC* inverts elastic weight consolidation (Kirkpatrick et al., 2017) by negating the quadratic penalty on the parameters with high donor-task Fisher, with a second penalty that keeps the weights near recent target-good values. *Anti-A-GEM* mirrors the A-GEM constraint (Chaudhry et al., 2018) by reflecting the component of the target update along the donor gradient whenever that update would also lower the donor loss. *Silent clone* copies target-selective MLP units over the least target-relevant donor-selective units in the late layers before continuing GRPO. None reaches the gain of the explicit LRA merge on non-math donors, and none approaches ANTIGEN.

E LOW-RANK ASCENT BY DONOR FAMILY: ALL SETTINGS

Tables 6 and 7 list every arm run from the Countdown specialist; Appendices B and D discuss them.

	Math			Non-Math Reasoning			Knowledge			Broad (Combined)		
Target Task	tgt	fgt	coll	tgt	fgt	coll	tgt	fgt	coll	tgt	fgt	coll
Countdown	−10.9	−10.3	−13.1	+3.5	−8.2[a]	−4.8	−2.7[c]	−10.8	−8.7	−21.3	−27.6[a]	−34.1
AI2-RC	−4.0	−11.9	−13.8	−4.0	n/m	−11.8	−2.5	−3.3	−0.7	−3.0	−18.4	−2.5
StrategyQA	−0.2	−38.5	−13.3	+0.4	−14.8	−5.3	−0.3	−9.2	−7.7	−1.6	−17.2	−9.2
StrategyQA (from base)	+0.4	−20.2	−10.2	−1.9	−12.9	−4.6	+1.5	−8.0	−6.8	−3.5	−12.5	−11.0
DeepScaleR	−7.6	−27.3	−9.4	−2.9	−12.9	−5.8	−1.8	−3.2	−2.0	−1.3	−10.5	−3.1
OpenBookQA	−3.1	−12.1	−10.2	−2.6	n/m	−7.1	+0.3	−5.9	−1.9	−4.8	−15.8	−4.6
Lawma (8B, SFT)	+0.2	−0.2	−1.9	+0.1	n/m	−2.3	+0.1	n/m	−0.8	+0.1	−5.8	−0.3
DAPO-17k	—	—	—	−0.3	−13.9	−4.0	—	—	—	—	—	—
FineProofs/MATH-500	+0.4	−4.7	−6.2	+1.4	−9.9	−5.3	+2.7	−2.8	−7.2	−2.7	−8.6	−12.9

Table 5: Full version of Table 2: every setting, including StrategyQA trained from the base model. Same columns and colouring; [a] forgetting measured against the specialist's own score at the start of the window where the control never scored the task, [c] run from the 1200-step Countdown specialist over steps 1220–1800 against the same control.

			Δ mean / Δ peak (points) over the window		
family	method	runs	1620–1800	1620–2000	1620–2400
Control	plain GRPO, 3 seeds (absolute mean / peak)	3	45.9 / 49.3	47.7 / 52.3	49.1 / 54.2
	plain GRPO, seed-42 re-run	1	-0.1 / -0.9	-1.2 / -1.7	–
ANTIGEN (ours)	ANTIGEN: broad-21 donors, fill ×25	4	+0.9 / +2.4$^{(4)}$	+2.4 / +5.0$^{(4)}$	+4.4 / +8.2$^{(3)}$
	ANTIGEN, ramped multiplier ×25	1	+2.1 / +3.5	+3.7 / +6.1	+5.0 / +6.2
	ANTIGEN, ramped multiplier ×40	1	+2.5 / +6.3	+0.6 / +5.3	–
	ANTIGEN, fill ×10	1	-0.6 / -0.3	+0.0 / +2.3	–
	ANTIGEN, non-math reasoning donors (2), ×10	1	+1.0 / +1.3	–	–
	ANTIGEN, knowledge donors (6), ×10	1	+1.1 / +3.5	–	–
	ANTIGEN, knowledge donors (6), ×10, vacate	2	-8.6 / -5.1$^{(2)}$	-8.2 / -7.7$^{(2)}$	–
	ANTIGEN + vacate 0.01, ×25	1	-3.3 / -2.7	–	–
	ANTIGEN, strong-ascent addresses (no ceiling, lr 1e-3, 100 steps)	3	+0.2 / +2.1$^{(3)}$	-2.3 / +2.2$^{(3)}$	-8.9 / -1.6$^{(2)}$
	ANTIGEN, math donors (2), ×25	1	+1.9 / +6.9	+2.9 / +5.5	–
	ANTIGEN, non-math reasoning donors (2), ×25	1	+2.0 / +4.3	-0.4 / +1.3	–
	LRE-addressed refill, ×25	1	+0.9 / +2.9	+0.7 / +1.7	–
	LRE-addressed refill, ×25, vacate	1	-2.0 / -1.9	–	–
Container ablations (ours)	random addresses r336, ×25	4	+1.1 / +1.8$^{(4)}$	+2.0 / +3.7$^{(4)}$	+3.0 / +5.3$^{(3)}$
	random addresses r1008, scale-matched	1	-3.4 / +0.1	-1.0 / +3.9	–
	full-rank container, ×25	1	-0.9 / -0.7	-2.9 / -2.3	–
	trainable A (plain LoRA) r336, ×25	4	+1.6 / +1.9$^{(4)}$	+2.9 / +8.1$^{(3)}$	+4.3 / +6.6$^{(3)}$
	trainable A (plain LoRA) r336, ×10	2	+0.6 / +1.4$^{(2)}$	+0.7 / +2.1	–
	trainable A (plain LoRA) r16, ×1.19	2	-1.0 / -1.7$^{(2)}$	-1.6 / -1.3	–
	layer-wise LR, layers 12–23 ×10	1	-3.5 / -1.1	-15.7 / -4.1	–
	layer-wise LR, layers 12–23 ×25	1	-36.2 / -7.3	–	–

Table 6: Countdown (Qwen3.5-0.8B): improvement over plain GRPO for every method run from the step-1600 anchor. Cells are Δmean / Δpeak countdown accuracy in points over the 3-seed control within a fixed window, so peaks are not length-biased; an arm is shown only where it covers the whole window. Per run we take the mean over the scored checkpoints and the peak checkpoint; over the runs of a recipe (seeds and same-recipe re-runs; superscript n) each statistic is averaged separately. The seed-42 control re-run shows the run-to-run noise floor. (a) ANTIGEN and its container ablations.

			Δ mean / Δ peak (points) over the window		
family	method	runs	1620–1800	1620–2000	1620–2400
Control	plain GRPO, 3 seeds (absolute mean / peak)	3	45.9 / 49.3	47.7 / 52.3	49.1 / 54.2
Forgetting methods	LRA, non-math reasoning (2), periodic	1	+6.7 / +5.3	+4.9 / +5.7	–
	LRA, non-math reasoning (3), periodic	1	+3.5 / +3.3	+0.0 / +0.3	–
	LRA, non-math reasoning (2), plateau	2	+0.4 / -0.6$^{(2)}$	-0.5 / -0.9$^{(2)}$	+1.8 / +0.6
	LRA, non-math reasoning (2), curriculum-scheduled	1	+5.6 / +4.7	+4.0 / +2.5	+2.1 / +0.6
	LRA, non-math reasoning (5), periodic	1	+1.8 / +4.1	-4.6 / +1.1	–
	LRA, non-math reasoning (5), plateau, normalized dose	1	-3.9 / -4.9	-5.3 / -7.9	–
	LRA, reasoning + MC (7), periodic	1	-1.5 / -0.7	-6.4 / -3.7	–
	LRA, math only (2), periodic	2	-10.0 / -7.0$^{(2)}$	-12.6 / -10.0$^{(2)}$	–
	LRA, knowledge (9), periodic	1	-7.7 / -8.5	-9.0 / -10.1	–
	LRA, math + knowledge (11), periodic	1	-10.8 / -8.3	-23.6 / -11.3	–
	LRA, math + knowledge w/o GSM8K (10), periodic	1	-21.4 / -19.3	–	–
	LRA, broad (10), plateau	1	-0.7 / -2.3	-2.9 / -5.3	-5.5 / -7.2
	LRA, broad (10), curriculum-scheduled	1	-0.1 / -1.7	-2.8 / -4.7	-7.2 / -6.6
	LRA, broad (10) + data curriculum	1	-2.7 / -2.7	-7.9 / -5.7	-26.5 / -7.6
	LRA, StrategyQA, plateau	1	-0.8 / -1.5	-1.2 / -2.1	–
	LRA, Dolly, plateau	1	-3.4 / -4.1	-4.8 / -6.5	–
	LRA, HotpotQA, plateau	1	-4.2 / -5.7	-5.3 / -6.5	–
	LRA, TriviaQA, plateau	2	-0.9 / -1.8$^{(2)}$	-2.6 / -3.3	–
	LRA, TriviaQA, plateau, raised dose	1	-0.1 / -1.3	–	–
	LRA, Alpaca + ELI5, plateau	1	-5.4 / -7.1	–	–
	LRA, Alpaca + ELI5, plateau, reduced dose	1	-3.7 / -3.9	–	–
	LRA, StrategyQA + TriviaQA, plateau	1	-0.7 / -2.5	-1.8 / -2.7	–
	LRA, non-math reasoning + TriviaQA (3), plateau	1	-1.5 / -2.7	–	–
	LRA, non-math reasoning + TriviaQA (3), plateau, normalized	1	-1.4 / -2.3	-2.7 / -3.7	–
	LRA, QASC + CSQA2, plateau	1	-3.3 / -4.1	-5.0 / -6.5	–
	LRE (extraction), tuned (6)	3	-1.0 / -1.3$^{(3)}$	-1.8 / -3.4$^{(3)}$	–
	LRE, second extraction	1	-6.0 / -7.1	–	–
	LRE, non-math reasoning (2), plateau	1	-30.9 / -9.1	-40.2 / -12.1	–
	anti-EWC (6)	1	+0.5 / +0.5	-0.5 / -1.7	–
	anti-EWC, tuned (6)	1	-1.2 / -1.9	-1.8 / -4.1	–
	anti-EWC, tuned, non-math reasoning (2)	1	-0.6 / -0.9	-0.2 / -0.5	–
	anti-EWC, late layers, non-math reasoning (2)	1	+0.3 / -0.3	-0.0 / -0.5	–
	anti-EWC, TriviaQA	1	-1.1 / -0.9	-2.0 / -2.9	–
	invasive gradient (anti-A-GEM) (6)	1	+0.3 / -1.1	-1.8 / -4.1	–
	invasive gradient, non-math reasoning (2)	1	-0.4 / -0.7	-2.3 / -3.7	–
	invasive gradient, TriviaQA	1	-2.1 / -3.3	–	–
	ascent–descent, layers 20–23 (6)	1	-3.2 / -3.1	-6.1 / -6.1	–
	ascent–descent, full network (6)	1	-4.6 / -4.5	-6.9 / -7.5	–
	ascent–descent, non-math reasoning (2), plateau	1	-0.2 / -0.3	-1.2 / -2.5	–
	silent clone, non-math reasoning donors	1	-3.5 / -3.5	–	–
	silent clone, knowledge donors	1	-8.3 / -8.3	–	–
	silent clone, knowledge donors v2	1	+3.4 / +3.9	+1.1 / +0.9	–
	silent clone, staged knowledge (8)	1	+0.9 / +1.9	-0.9 / -1.1	–
	silent clone, staged, double dose	1	-1.3 / +1.9	-1.2 / +2.3	–
	silent clone, staged, plateau-gated	1	+1.7 / +2.9	-2.0 / -0.1	-5.6 / -2.0
	target-exclude r32	1	-2.3 / +2.5	–	–
	target-exclude r256	1	-3.3 / +2.9	–	–
	weight decay (blanket forgetting)	1	-2.4 / -4.3	–	–
RL plasticity baselines	ProRL: reference reset every 100 steps	1	+0.7 / +0.7	-0.6 / -2.3	–
	ProRL: reference + optimizer reset on plateau	1	+0.3 / -1.7	+0.2 / +1.5	+1.0 / +2.2
	Clip-Cov	2	+0.3 / -0.8$^{(2)}$	-1.5 / -3.5	-1.6 / -1.8
	top-20% entropy tokens	2	+0.3 / -1.1$^{(2)}$	+0.8 / -0.9	+0.5 / -0.8
	KL replay, λ=1	2	-1.4 / -2.4$^{(2)}$	-3.0 / -5.3	-4.3 / -5.4
	KL replay, λ=10	1	-1.9 / -3.9	–	–
	ReDo, 5% dormant units	1	-24.6 / -22.1	–	–
Data baselines	difficulty curriculum	1	-1.2 / -2.3	-2.7 / -3.7	-3.8 / -5.6
	step curriculum	1	+0.0 / -1.1	-2.0 / -4.1	-2.4 / -2.0

Table 7: Countdown (Qwen3.5-0.8B): improvement over plain GRPO for every method run from the step-1600 anchor. Cells are Δmean / Δpeak countdown accuracy in points over the 3-seed control within a fixed window, so peaks are not length-biased; an arm is shown only where it covers the whole window. Per run we take the mean over the scored checkpoints and the peak checkpoint; over the runs of a recipe (seeds and same-recipe re-runs; superscript n) each statistic is averaged separately. The seed-42 control re-run shows the run-to-run noise floor. (b) Forgetting methods, RL plasticity and data baselines.

F EXPERIMENTAL SETTINGS AND REPRODUCIBILITY

Models. All GRPO experiments use Qwen3.5-0.8B (Qwen Team, 2026) in non-thinking mode, with Qwen3.5-2B for the DeepScaleR replication. The SFT experiments on FineProofs use the same 0.8B model, and the Lawma setting of Table 2 uses Llama-3-8B.

GRPO. We use TRL's GRPO trainer (Shao et al., 2024) with the recipe of QED-Nano (Qu et al., 2026): learning rate 10^{-6}, constant after a 3% warm-up, no KL penalty ($\beta = 0$), sampling temperature 0.8, eight generations per prompt and 48 completions per optimizer step (six prompts), bf16 weights and rollouts generated by a colocated vLLM engine. Completions are capped at 1,024 tokens for Countdown, DeepScaleR and DAPO-17k and at 512 tokens for OpenBookQA, AI2-RC and StrategyQA. Checkpoints are saved and scored every 20 steps on Countdown and every 40 steps elsewhere. The Countdown specialist θ_S is the checkpoint after 1600 steps of this recipe from the base model; the DeepScaleR, StrategyQA and AI2-RC specialists are taken at 600, 600 and 1600 steps, and the OpenBookQA specialist at 1200 steps. DeepScaleR training uses the subset of problems on which the base model's pass@8 lies in $(0, 0.75]$, 8,523 problems, since three quarters of the raw set produce no reward at 0.8B. All arms are continuations of the specialist under the same recipe; resuming or re-initializing the optimizer state at the anchor made no measurable difference for plain GRPO, so we do not distinguish the two.

Saturation. The anchor at step 1600 was chosen where the improvement rate had collapsed: the run gained 2.7 points per 100 steps on average from the base model, 1.2 over steps 800 to 1600 and 0.1 over steps 1200 to 1600, reaching 44.6% at the anchor. The continued control is not flat afterwards. Over three seeds it climbs at 0.7 ± 0.5 points per 100 steps between steps 1620 and 2400, with window means of 45.9, 47.7 and 49.1% over 1620 to 1800, 2000 and 2400 and a 54.2% peak, which is why every comparison in the paper is made against the continued control over matched windows rather than against the anchor. The DeepScaleR control climbs at 0.5 points per 100 steps over the compared window (steps 600 to 1200) after 1.0 before it, and the OpenBookQA controls at 1.5 to 1.8 points per 100 steps over their first 600 steps and 0.1 to 0.3 over steps 600 to 3000 (three seeds each).

Seeds and windows. The headline Countdown comparison uses three seeds (42, 43, 44) for both GRPO and ANTIGEN, each seed changing the data order and the rollout sampling; random-address adapters use an additional fixed address seed. For each arm we report the mean over the scored checkpoints in a fixed window and the peak checkpoint, computed per seed and then averaged separately across seeds, so that peaks are not biased by run length. The main-text figures show the mean across seeds at every scored step with a band of one sample standard deviation across seeds. The RL plasticity baselines were screened with one seed for 200 or 400 steps from θ_S, and the strongest of them (ProRL on plateau, Clip-Cov, high-entropy tokens) were extended to 2400 steps with seeds 42 and 43; Tables 6 and 7 mark the number of runs behind each cell. Each DeepScaleR and OpenBookQA experiment uses the same three seeds (42, 43 and 44) for both the control and the from-base ANTIGEN arm, at 0.8B and again at 2B; the arm attached to the specialist (Appendix B) and the 38-task evaluation of Appendix B.1 use seed 43 alone.

Evaluation. Generation during evaluation uses temperature 0.7, top-p 0.8, top-k 20 and no minimum probability, following the model card's recommendation for non-thinking mode. Every task is capped at 500 items with a fixed seed. Multiple-choice probes are scored by likelihood over the options, generative tasks by exact match after answer extraction. The target task is scored at every saved checkpoint and the forgetting probes at every fourth one, and the window statistics in the tables average over the checkpoints at which each task was scored.

Compute. Experiments ran on a single node with eight NVIDIA A100-SXM4-80GB GPUs. A GRPO run uses two GPUs and takes about 43 seconds per optimizer step at 48 completions, so a 400-step continuation takes about five hours plus roughly one hour of evaluation, and a 1200-step DeepScaleR run about fifteen hours. The step time is bound by the decode latency of the longest completion and does not shorten with more GPUs. Software versions were PyTorch 2.11 (CUDA 12.8), Transformers 5.13, TRL 1.7.1, PEFT 0.19 and vLLM 0.24. Code, configurations for every run in the tables and the evaluation outputs will be released.