Fine Until Fine-Tuned: Repeated Solutions Make Reasoning Fragile

Anonymous authors
Paper under double-blind review

Abstract

Recipes such as s1 and LIMO teach a model to reason with little data by showing it the same thousand or fewer worked solutions many times over. Judged when that training ends, the repetition looks harmless. But reasoning models are often trained again, and we find that repetition leaves their reasoning fragile to that next stage, even when the stage has nothing to do with reasoning. We fine-tuned Qwen3.5-9B-Base on its own correct solutions to competition math problems, either drilling a few hundred of them about eight times each or showing many more once; with the same amount of training, both solve about 95% of held-out problems. A single pass of ordinary instruction tuning leaves the once-trained model where it was, while the drilled one falls to 86.0%, and harsher later stages take it to 59.3% or below. A third model that visited the drilled problems just as often, with a new solution at every visit, was unharmed, so the damage comes from seeing the same texts again rather than from having few problems. The break recurs with a stronger model's traces, in further training runs and on other models and tasks. It is also cheap to undo: the reasoning is suppressed rather than erased, and five updates of reasoning training bring almost all of it back, as does brief training on the reasoning format with almost no mathematics. Fresh solutions prevented the damage, and so did replaying 6.25% of the original solutions in a gentler later stage, so our claim concerns later training without such replay. Sharpening alone does not explain the break, since a model sharpened three-quarters as much without repetition was unharmed, and on a skill the base model could not perform within a token budget, repetition mainly cost learning.

1 Introduction

Teaching a language model to reason can take surprisingly little data. Recipes such as s1 and LIMO fine-tune a model on a thousand worked solutions or fewer, problems written out step by step down to the answer, and make up for the small number by going over them again and again: s1 trains on its set 5 times and LIMO 15 times (Muennighoff et al., 2025; Ye et al., 2025). Evaluated at handoff, as we will call the point where this training ends, the repetition looks free and sometimes helpful: Kopiczko et al. (2026) found that 400 examples repeated many times over beat 128 times as many examples seen once.

Handoff, though, is rarely the end of a model's training. A released reasoning model is often fine-tuned again, by its developers or by whoever adapts it next, and the new training need not have anything to do with reasoning. How much of the reasoning survives that step is something a handoff score cannot tell. We wanted to know whether repeating the same solutions many times makes the reasoning more fragile to that next stage.

The comparison we set up changes only the repetition. We start from Qwen3.5-9B-Base, which already solves 94.7% of our 221 held-out competition problems from MATH-500 when it reasons. (Throughout, we open the model's reasoning block for it, so that it always reasons before answering, and call the result forced accuracy.) We then train it on its own correct solutions to other competition problems. These teach it nothing new, which is the point: whatever later separates the trained models must come from how the solutions were presented. The drilled model sees 579 solutions, one per problem, about eight times each. The once-trained model sees 4,480 different solutions once each, in

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.

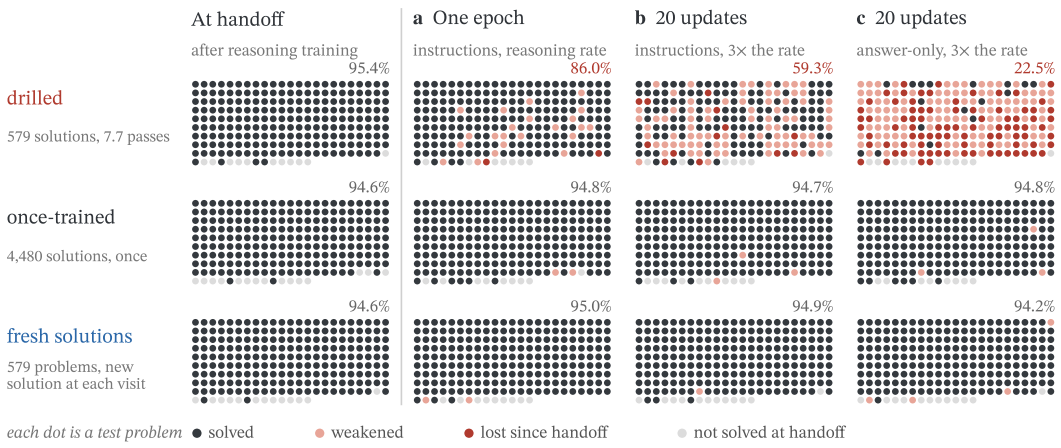

Figure 1: Same start, different end. Each dot is one of the 221 held-out MATH-500 problems, in the same place in every grid (easiest first); dark: solved in at least 3 of 4 samples. At handoff the three 9B models solve the same problems. After a later stage that does not train them to solve math problems, the once-trained and fresh-solution models still do, while the drilled model loses many (red: solved at handoff and in none of 4 samples after; light red: in 1 or 2), the more so the harsher the stage: (a) one epoch of instructions at the reasoning rate; (b) 20 updates of instructions and (c) 20 of answer-only training at three times that rate.

exactly as many updates. At handoff their accuracy is the same: both solve about 95% of the held-out problems, as the base model does.

Then we train both a little further on something unrelated. A single pass over 8,544 everyday instructions, one epoch at the learning rate of the reasoning training, is the kind of stage a released model might go through next. The once-trained model comes out of it where it went in. The drilled model falls from 95.4% to 86.0%, losing 9.6 points more than the once-trained model (Figure 1a), and the harsher the later stage, the further it falls. In two stress tests at three times the learning rate, twenty updates of the same instructions leave it at 59.3%, and twenty that reward bare answers leave it at 22.5% (Figures 1b and 1c), while the once-trained model loses about three points at most.

Two things set the drilled model apart: it saw each solution many times, and it saw far fewer problems. A third model, the fresh-solution model, separates the two. It visits the drilled model's 579 problems just as often, but with a different correct solution at every visit, so it has as few problems as the drilled model yet sees each text about once. It came through every later stage we gave it unharmed. The damage therefore comes from seeing the same solution texts again, not from having few problems; this was the test of the cause we had registered in advance.

Nor is the break a quirk of our setup. On long traces written by a stronger model, gpt-oss-120b, which is how s1 and LIMO are actually trained, the drilled model broke even further in the stress tests (Table 1), and fresh traces again prevented it. It broke again in two more training runs with new problems, orders and initializations, with larger adapters, at 35B, in Nemotron-3 Nano from another model family, and on an arithmetic search task (Figure 2). How much it breaks varies a great deal from run to run, so we claim the direction of the effect and not its size.

What the later stage takes away turns out to be cheap to give back (Section 4). Opening the drilled model's reasoning block does not bring its accuracy back, and nudging it with "Wait" whenever it stops early wins back less than half of what it lost. A few updates of training do far better. Five updates of reasoning training on new problems, 160 examples in all, bring it back to 93.9%, next to the once-trained model's 94.7%, and twenty updates on the base model's answers to chat prompts from non-math sources, which teach the format of reasoning rather than mathematics, do the same. The later stage suppresses the drilled model's reasoning rather than erasing it, although for anyone who fine-tunes such a model and stops there, the result is the same.

Could the damage have been seen coming? Accuracy at handoff gives no hint, but the drilled model already finds the base model's own solutions about 10% less likely per token, because repetition

makes the base model's most probable choices more certain still (Section 5). This over-sharpening is the natural suspect, yet a model we sharpened three-quarters as much without any repetition came through both harsh stages unharmed. The repeated texts matter beyond the sharpening they cause.

The fixes we tested are simple (Section 6). Sampling a new correct solution each time a problem comes around prevented the damage wherever we tried it. Mixing 6.25% of the original solutions back into a gentler later stage prevented it there too, so our claim concerns later training that does not replay the reasoning data, as when someone adapts a released model to their own task. It also concerns training that reshapes a skill the base model already has, which is what s1 and LIMO set out to do: on a synthetic skill the base model could not perform within a fixed token budget, the clearest cost of repetition was that the model learned less.

2 Setup

Models. Most experiments use Qwen3.5-9B-Base (Qwen Team, 2026), trained and sampled through Tinker (Thinking Machines Lab, 2025) with rank-32 LoRA adapters (Hu et al., 2022). To make sure the effect does not belong to one model, we repeated the main design on a larger model from the same family, Qwen3.5-35B-A3B-Base, and on Nemotron-3 Nano (NVIDIA, 2025), a hybrid Mamba–Transformer from another family that has already been post-trained. Both are mixtures of experts with about 3B active parameters.

Three ways to train on the same solutions. The training texts are the base model's own correct solutions to competition problems from NuminaMath-TIR, as included in the Tülu 3 mixture (Li et al., 2024; Lambert et al., 2024), sampled with the reasoning block open (Appendix L). Every model gets 140 updates of 32 examples at learning rate 10^{-4}; only the texts differ. The drilled model cycles through 579 solutions, one per problem and a random subset of the once-trained model's 4,480, so it sees each one 7.7 times on average, and we also keep it after 1.1 and 3.3 passes. The fresh-solution model visits the drilled model's problems in the same order and as often, drawing a different correct sample at each visit from 8 kept per problem, so it too sees each text about once. Around this core we trained the three models on traces that gpt-oss-120b (OpenAI, 2025) wrote for problems from the same source, repeated the design on the model's own solutions twice with new drilled problems, orders and adapter seeds, drilled a second set of 579 problems, and repeated the drilled run with rank-128 adapters (Table 2).

The later stages. After the reasoning training, each model and the untrained base get a second round of training on something else, with a fresh optimizer; the base starts it from a new adapter and each trained model continues its own. None of these stages teaches the model to solve math problems. The realistic one is a single epoch of *instruction tuning* on human-written instructions from No Robots (Rajani et al., 2023): one pass over 8,544 examples in 267 updates at 10^{-4}, the learning rate of the reasoning training. The stress tests are shorter and harsher, twenty updates at 3×10^{-4}, three times that rate, either on 640 of the same instructions or in the *answer-only* stage. That stage trains on short program-synthesis puzzles, such as finding a program of modular arithmetic instructions that turns a start value into a target, with bare answers as targets, so it rewards answering without thinking first. Section 3.2 also varies the rate and length and tries a broad chat mix.

How we measure. The test set is the 221 problems of MATH-500 at difficulty levels 3 to 5 that have integer answers (Hendrycks et al., 2021; Lightman et al., 2024), with 59 problems from AIME 2025 and 2026 as a harder check (Balunović et al., 2025). We sample four answers per problem, each with up to 16,384 tokens, and report *forced* accuracy, where the response starts with `\n<think>\n`, the opening of the reasoning block; Section 4 compares it with *free* accuracy, where the model completes the prompt however it likes. A model's *extra loss* is how much more forced accuracy it lost in a later stage than a comparison model, each measured from its own handoff. Intervals are 95% paired bootstrap intervals over problems. They do not cover training runs, so we repeated the 9B design.

Registered predictions. Before each experiment we registered its design, our predictions with numerical thresholds, and what each possible result would mean for our claims. Appendix O lists all 70 predictions, including the 17 that failed, 12 of them on the arithmetic task.

Table 1: Forced accuracy (%) of the 9B models on MATH-500 (levels 3–5) at handoff and after each later stage. Passes: how many times the average training text was seen. One epoch: instruction tuning at 10^{-4}; stress tests: twenty updates at 3×10^{-4}. The sharpened model is trained like the once-trained one on solutions sampled at temperature 0.5 (Section 5). Dashes mark stages we did not run. Table 3 adds free accuracy, B, the other models and the arithmetic task.

					Stress tests	
Training texts	Model	Passes	Handoff	One epoch	Instructions	Answer-only
None	Base	–	94.7	94.5	95.6	91.7
Own solutions	Drilled	1.1	95.4	–	94.5	94.3
	Drilled	3.3	95.1	–	94.0	92.8
	Drilled	7.7	95.4	86.0	59.3	22.5
	Once-trained	1.0	94.6	94.8	94.7	94.8
	Fresh solutions	1.0	94.6	95.0	94.9	94.2
	Sharpened	1.0	93.1	–	95.0	92.4
gpt-oss-120b traces	Drilled	7.8	95.0	–	33.7	1.1
	Once-trained	1.0	96.3	–	94.6	94.8
	Fresh traces	1.0	96.0	–	95.4	95.4

3 Repeated solutions make reasoning fragile

3.1 Handoff and one epoch of instruction tuning

Table 1 puts every 9B model side by side, at handoff and after each later stage (Table 8 gives the intervals). At handoff the rows barely differ. The 9B models trained on the base model's solutions in the first run, the second drilled set and the larger adapters included, solve between 93.9% and 95.6% of the test problems, next to the base model's 94.7%; only the sharpened model of Section 5 sits lower, at 93.1%. We had registered that the drilled and once-trained models would count as equally accurate if they differed by less than 5 points, and the largest difference in any math setting was 1.8.

The first later stage is the realistic one, a single epoch of instruction tuning at the learning rate the models were trained to reason with. The base, the once-trained and the fresh-solution models all end within a point of where they started. The drilled model loses 9.4 points, 9.6 more than the once-trained model (interval [6.9, 12.4]). Both halves of this were registered predictions: an excess of at least 5 points with its interval above zero, and a fresh-solution model within 3 points of the once-trained one, which came to −0.2.

3.2 Harsher and broader later stages

The stress tests push much harder, and the drilled model gives way. Instruction tuning at three times the rate costs it 36.2 points more than the once-trained model, and the answer-only stage 73.1 more (intervals [31.9, 40.4] and [69.3, 76.7]). No other model trained on the base model's solutions loses more than about three points, not even the drilled checkpoints saved after one and three passes (the 35B model and Nemotron broke sooner; Appendix B). On AIME, instruction tuning takes the drilled model from 69.9% to 22.0% and the once-trained one from 69.5% to 70.8%.

What sets the size of the break is mostly the learning rate of the later stage (Figure 2a). At the reasoning training's own rate the extra loss stays at about 10 to 13 points however long the stage runs: 12.6 after twenty updates on the instructions, 13.1 after sixty, which match the stress test's learning rate times updates, and 9.6 after the full epoch, with 10.3 for a broad stage of chat examples from the non-math sources of Tülu 3. At three times the rate it is 36.2 after twenty updates and 55.2 after sixty, and the drilled model does not recover along the way (Appendix E). The once-trained models stay within two points of their handoff accuracy in every one of these runs.

3.3 Repeated texts or few problems?

The fresh-solution model was built to tell repeated texts from few problems. It trains on exactly the drilled model's 579 problems, in the same order and just as often, but gets a new correct solution

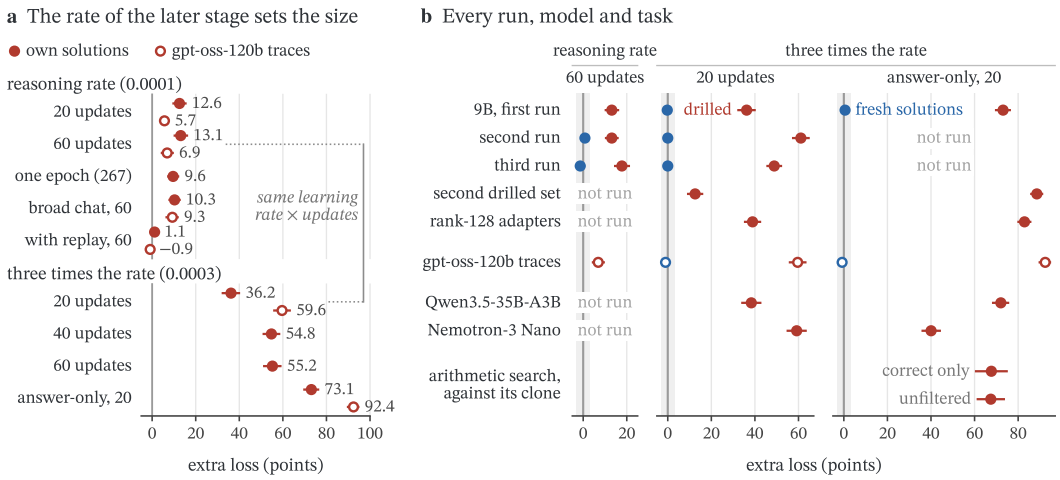

Figure 2: The drilled model's extra loss over the once-trained model, in points with 95% intervals, on one scale. (a) The 9B model after each later stage, instruction tuning unless named; filled: own solutions, hollow: gpt-oss-120b's traces. Sixty updates at the reasoning rate have the learning rate times updates of twenty at three times it; replay mixes 6.25% of the model's own training texts into each batch. (b) Every run and model, and the arithmetic task against a clone of its teacher; blue: fresh solutions; grey: the registered margin of ±3 points.

at each visit, so if having few problems were the trouble, it would break too. It does not. After the realistic epoch its excess over the once-trained model is −0.2 points, and after both stress tests it stays within a point, with intervals that include zero, while the drilled model loses 36.4 and 72.5 points more than it (intervals [32.4, 40.4] and [68.8, 76.0]). This was our registered test of the cause: the repeated texts would count as the cause if the drilled model lost at least 20 points more than the fresh-solution model in each stress test, with intervals above zero. Both passed.

3.4 A stronger model's traces

Neither s1 nor LIMO trains on the model's own solutions; both repeat long traces written by a stronger model. So we trained the same three models on traces from gpt-oss-120b. Once again the three are level at handoff, and once again only the drilled model breaks in the stress tests, this time by more: 59.6 points beyond the once-trained model under instruction tuning and 92.4 in the answer-only stage, which leaves it almost no correct answers (1.1%). As we had predicted, a fresh trace at every visit prevents it. At the reasoning training's own rate the order reverses: after sixty updates the drilled model has lost 6.9 points more than the once-trained model, less than the 13.1 with its own solutions and short of the 10 we had predicted, so that prediction failed. Training on these traces also moved the model only a little toward gpt-oss-120b, so the experiment tests the repetition of a stronger model's texts rather than the full fine-tuning of s1 and LIMO (Section 8).

3.5 Other runs, other models and an arithmetic task

Because our intervals cover test problems and not training runs, we ran the 9B design twice more with new drilled problems, orders and adapter seeds, and tested each run after the instruction stress test and after sixty updates at the reasoning rate. In both new runs and both stages the drilled model again lost more than the once-trained and the fresh-solution models, with every interval above zero, as we had registered (Appendix N). What changed was the size. In the stress test its extra loss was 36.2 points in the first run and 61.1 and 48.9 in the new ones, and 12.6 for the second drilled set of the first run; at the gentler rate the runs agreed better, at 13.1, 13.1 and 17.8. The fresh-solution models stayed within 1.4 points of the once-trained ones throughout.

Figure 2b gathers every run, model and task, and in each one the drilled model's interval lies above zero, whatever its size. The 35B model needs one remark. Its instruction stress test is hard even on the untrained base, which starts from a new adapter and falls from 96.0% to 48.1%, so there we compare

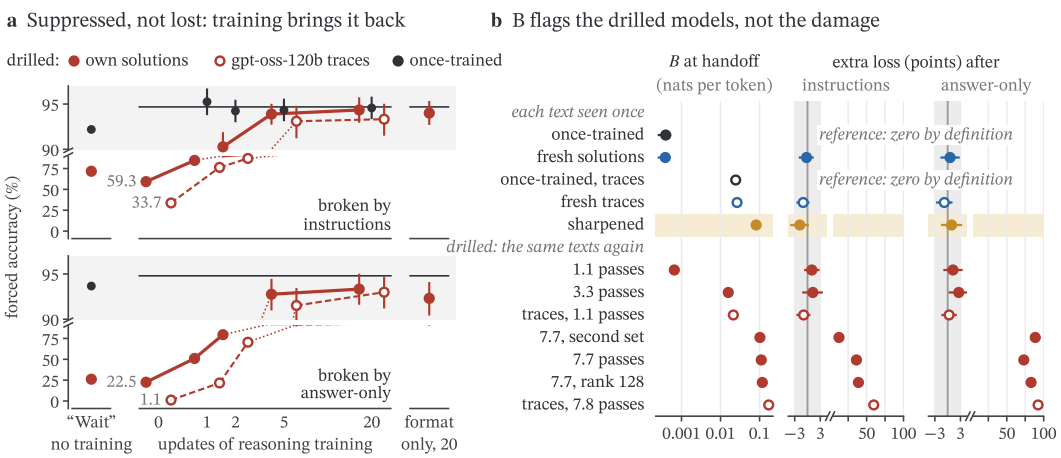

Figure 3: (a) Forced accuracy after k updates of reasoning training for the drilled models broken by each stress test (filled: own solutions; hollow: gpt-oss-120b's traces), against the once-trained model after the same stress test (line) and trained the same way (dots). The grey strip magnifies 90–97%; bars are the paired 95% intervals of R_k, the share of the gap closed. "Wait": both models nudged without training; format only: twenty updates on chat responses that show only the format of reasoning. (b) Each 9B model's divergence B at handoff (log scale) and extra loss after each stress test, with 95% intervals; grey: the registered margin of ±3 points.

only the trained models: the once-trained model goes from 96.7% to 91.0%, while the drilled one falls to 50.8% (Appendix B). On an arithmetic search task, a variant of Countdown (Gandhi et al., 2024), two models drilled on an RL teacher's samples lose 69.5 and 69.8 points in the answer-only stage, while the teacher, a clone trained on its samples about once each, and the starting model lose between 2.1 and 4.2 (Appendix H).

4 Suppressed, not lost

There are two ways to lose accuracy after later training: a model can stop choosing to reason, or it can reason worse. Instruction tuning does the first to every 9B model of our first experiment. Left to answer freely, none of them opens its reasoning block any more, and free accuracy falls to 22.4–33.5%, the base included. Measured that way, the drilled and once-trained models look only 9.7 points apart, against 35.4 with the block opened, so the lost habit hides most of the difference. This is why we force the reasoning (Appendix D).

Forcing the block open does not settle the matter either, because a model can still give up partway, and the drilled models often do, stopping after a few lines or going round in loops (Appendix C). After the answer-only stage, for example, the drilled model's median response is 103 tokens long, against 1,294 for the once-trained model. So we pushed it on: whenever a response ended before 1,024 tokens or without a boxed answer, we appended "Wait" and let the model continue, up to 8 times, as in the budget forcing of Muennighoff et al. (2025). This won back 41% of the drilled model's extra loss after instruction tuning and 6.5% after the answer-only stage (Table 6). We had registered this outcome as a sign of lost competence (RF1), a reading the next experiment supersedes.

Training worked where nudging had not. We continued training each broken drilled model at the learning rate of the reasoning training, on the base model's own correct solutions to 640 problems that no model had seen, and measured forced accuracy after 1, 2, 5 and 20 updates. To compare the models we use R_k, the share of the gap to the once-trained model that k updates have closed, which runs from 0 at the start to 1 when the drilled model has caught up. After the instruction stress test, a single update of 32 examples took the drilled model from 59.3% to 84.7%, and five took it to 93.9%, or $R_5 = 0.98$ (interval [0.94, 1.01]). The other three broken models, one of them starting from 1.1%, reached R_5 between 0.96 and 0.98 (Figure 3a), while the once-trained model, trained the same way,

stayed within 0.6 points of where it was. We had predicted R_5 of at least 0.5, and fixed in advance that 0.75 or more would mean the drilled model's reasoning was suppressed rather than lost.

Training on mathematics could still be teaching mathematics. So we also trained the two drilled models broken on their own solutions with the base model's forced responses to 640 prompts from the broad chat mix. These show the format of reasoning, opening the block, reasoning and closing it before answering, with almost no mathematics in them (Appendix N). Twenty updates brought the drilled model back to 94.0% after instruction tuning and 92.3% after the answer-only stage, R_{20} of 0.98 and 0.97. We had registered that a share of at least 0.5 here would make us drop the claim that the loss is more than a lost habit, and we drop it.

The later stage, then, suppresses the drilled model's reasoning in a way that neither opening the block nor nudging undoes, but that a few updates of training on how to reason do, much as fine-tuning is known to wrap capabilities rather than remove them (Jain et al., 2024). With LoRA the base weights never change, so some recovery was possible in principle; the experiment shows how little it takes. For anyone who fine-tunes a drilled model and stops there, the reasoning is gone in practice.

5 What repetition changes at handoff

If accuracy at handoff cannot tell the drilled model from the others, something else might. A natural place to look is how each trained model treats the base model's own reasoning. For a trained model θ we take solutions that the base model wrote with its reasoning forced, and measure how much less likely they are under θ than under the base model, averaged over the positions P after the prefix:

$$B(\theta) = \frac{1}{|P|} \sum_{t \in P} \Big[\log p_{\text{base}}(y_t \mid y_{<t}) - \log p_\theta(y_t \mid y_{<t}) \Big].$$

B is zero if training has left the base model's reasoning untouched and grows as the trained model moves away from it. We draw the samples at temperature 1 and cut them at 2,048 tokens, so B estimates the forward divergence $\mathrm{KL}(p_{\text{base}} \,\|\, p_\theta)$, which Shenfeld et al. (2026) relate to forgetting.

B does pick out the drilled model (Figure 3b). Under the drilled 9B model the base model's solutions lose 0.111 nats per token, which makes a token about 10% less likely on average, and the loss grows with the number of passes. Under the fresh-solution model they lose 0.0004, as little as under the once-trained model, so the high B comes from the repeated texts. For scale, training once on gpt-oss-120b's traces, which come from a different model altogether, moves the 9B model only 0.024. The drilled model stands out in the same way at 35B, on Nemotron, on the second set of problems (Table 3) and on gpt-oss-120b's traces (0.171 against 0.027 for fresh traces).

Most of this change is of one kind. Split by position, 63% of the drilled model's B comes from places where it still picks the base model's top token but is even more sure of it, taking probability from the alternatives the base model would sometimes choose. We call this over-sharpening. Sharpening a model toward its own likely outputs is how self-training and RL are thought to improve reasoning (Huang et al., 2025; Yue et al., 2025), and repetition pushes it much further. It is still far from memorization: the drilled model's top next token matches 90.7% of the tokens in its training texts, against 85.1% for the base model.

Over-sharpening is the obvious suspect for the damage, so we tested it directly, with a model sharpened without repetition. It trains exactly like the once-trained model, on the same problems seen about once each, but each solution is the base model's first correct sample at temperature 0.5 instead of 1 (Appendix N). Samples drawn at a low temperature concentrate on the base model's most likely choices, so training on them should sharpen the model toward those choices without repeating any text. The model moved well away from its base: its B was 0.082 (interval [0.077, 0.087]), three-quarters of the drilled model's 0.111, against 0.0004 for the once-trained model. Yet it was not fragile. After the two stress tests its excess over the once-trained model was −1.8 and 0.9 points (intervals [−4.1, 0.3] and [−1.6, 3.4]), where the drilled model's were 36.2 and 73.1. We had registered that a result within 3 points of the once-trained model would mean that seeing the same texts again matters beyond the sharpening it causes, and that is our reading, with two caveats. We did not check that the sharpened model's divergence falls where the drilled model's does, and since the drilled model went from unharmed at 3.3 passes (B of 0.016) to broken at 7.7, a threshold between the sharpened model's level and the drilled one's is not excluded.

B therefore flags drilled models without identifying the mechanism. It catches every drilled model before any later training, at the cost of one forward pass per base sample, but it says nothing about how much a model will lose. Nor does its level decide the outcome on its own: a high B without repetition did no harm, and an exploratory arithmetic model that mixed samples from its starting model into its drilling kept B at 0.0085 and still lost 27.1 points (Appendix G).

6 Two remedies, and where the claim stops

Replay. The standard defence against forgetting in continual learning is to mix some of the earlier data into later training (Rolnick et al., 2019; Scialom et al., 2022). We tried a small dose of it. In sixty updates of instruction tuning at the reasoning training's rate, two of every 32 examples were the model's own training texts, 6.25% of each batch, and the break disappeared: the drilled model's extra loss was 1.1 points (interval [−0.9, 3.3]) against 13.1 without replay, and with gpt-oss-120b's traces −0.9 ([−2.8, 0.8]) against 6.9. We had predicted that replay would at least halve the break, and had decided in advance that a stronger result would limit our claim to later training without replay, as in a downstream user's fine-tuning. We did not test replay in the harsher stages.

Fresh solutions. The remedy our design was built around also held up. A new correct solution at every visit prevented the damage in every stage we ran it through: the realistic epoch, both stress tests and the gentler stage of the two new runs, and, with gpt-oss-120b's traces, the stress tests (Section 3). It needs 8 correct solutions per problem instead of one, but no more problems. On the 9B model, where there was little accuracy to gain, it cost none; on the synthetic skill below it did cost some.

A skill the base model lacks within a budget. Everything so far reshapes a skill the base model already has. That is the setting s1 and LIMO aim at, since both start from the premise that the base model knows enough and needs only to be shown how to use it (Muennighoff et al., 2025; Ye et al., 2025), but it leaves open what happens when training teaches something new. Our base model solves most competition problems we could draw, so we built a task instead: multiplying two five-digit numbers in base 7 within 2,048 tokens. Given 8,192 tokens the base model gets 75.0% of a small sample right, but within 2,048 it solves 0.2%, while a solver's worked solutions average 871 tokens, so what training teaches here is an efficient procedure the base model lacks. We fixed the budget before sampling the base model at it, trained the three models as in mathematics, and tested them after sixty updates of instruction tuning at the reasoning rate and after the instruction stress test.

Here the clearest cost of repetition was in learning. At handoff the once-trained model solved 66.3% of the held-out problems, the fresh-solution model 58.1% and the drilled model only 37.9%. Our registered criterion compares accuracy after the gentler stage, and by it the drilled model breaks, ending 24.2 points below the once-trained model, but that gap was there at handoff. Measured from each model's own handoff, a reading we added after the runs, the drilled model lost no more points than the once-trained one (an extra loss of −4.2, interval [−9.3, 0.9]), though as a share of what it had learned it lost somewhat more (29.9% against 23.4%), and the fresh-solution model the most (33.4%). The stress test wiped out the skill's accuracy in all three. The large fragility we found thus belongs to repetition when training reshapes a skill the base model already has. On a procedure new to it, repetition's main cost was in learning, and what was learned was fragile whatever the recipe. With one synthetic task, this marks one boundary rather than mapping it.

7 Related work

Repetition and durability. Kopiczko et al. (2026) show that for long chain-of-thought fine-tuning, many epochs on a small set beat one epoch on far more data, measured right after training; whether that advantage lasts is the question we take up. Outside reasoning, repeated text is already known to make learning less durable. A fact seen in identical text is forgotten faster than one seen in paraphrases (Chang et al., 2024), edits trained with paraphrases survive later fine-tuning better (Wen & Zhang, 2025), and in pretraining, data repeated many times harm a model out of proportion to their share of the training set (Hernandez et al., 2022), most at an intermediate number of repeats (Chudnovsky et al., 2026). Closest to our setting, more epochs of reasoning fine-tuning can win after training and lose after a later RL stage that trains the same skill (Kang et al., 2026). Our design isolates the repetition: the number of updates, the source of the solutions and the accuracy at handoff are fixed,

the fresh-solution model plays the role of the paraphrases, and the later stages do not train the skill at all. And because the drilled model ends well below the base model it started from, the later stage suppresses reasoning the model had, rather than merely adding less to it.

What makes a skill last. Shenfeld et al. (2026) find that forgetting of earlier capabilities tracks the forward KL from the base model, measured on the new task. Models also forget less when their training data are regenerated at each epoch rather than reused (Chen et al., 2026), and when they train on their own rewrites of the targets (Yang et al., 2024). These studies follow old skills while a new one is learned; we follow the newly trained skill through the next stage. How a capability is acquired also shapes how well it is retained, in ways that performance right after training does not reveal (Feng et al., 2026; Springer et al., 2025). B is a divergence of the kind Shenfeld et al. study, and our sharpened model shows that a large one need not make a skill fragile.

Losses that can be recovered, and sharpening. Capabilities that seem lost after fine-tuning are often still there and can be brought back (Kotha et al., 2024; Jain et al., 2024; Zheng et al., 2025; Qi et al., 2025). Knowledge removed by unlearning returns after brief fine-tuning on loosely related data (Hu et al., 2025), much as our drilled models' mathematics returns after brief training on chat responses from non-math sources, and fine-tuning on responses without reasoning traces suppresses explicit reasoning while accuracy given valid reasoning stays high (Twist et al., 2026). On the sharpening side, RL mostly reweights reasoning the base model already produces (Yue et al., 2025; Venhoff et al., 2026; Huang et al., 2025), and longer fine-tuning raises perplexity on reference solutions while accuracy holds (Ruan et al., 2025) and lowers pass@k (Dang et al., 2025; Nguyen et al., 2026). Qwen base models already verify and backtrack (Gandhi et al., 2025), and our results may depend on this.

8 Discussion and limitations

For people who train reasoning models, the practical lesson is to test a model after the next stage of its training, with its reasoning forced as well as free: in none of our math runs did accuracy at handoff tell the drilled model apart. If a recipe must revisit the same problems, fresh solutions prevented the damage wherever we tried them, replay did so in the one stage we tried, and about 160 examples of reasoning training repaired it after the fact.

How the repetition does its damage is still open. Over-sharpening seems not to be enough, since a model sharpened three-quarters as much without repetition was not fragile, though a threshold between its level and the drilled one's is not excluded. What the drilled model has and the sharpened one lacks is a close fit to a few hundred particular texts (0.277 nats per token, against the base model's 0.453). Another untested account is that drilling leaves the adapter in a sharper region of parameter space, where any later update does more damage (compare Springer et al., 2025); a jump in the drilled models' training loss at the first update of the arithmetic task's answer-only stage would fit it (Appendix I). The speed of relearning fits either account: the later stage seems to push the drilled model a short way off a narrow path, and a short way back is enough (see also Appendix F).

Limitations. All training uses LoRA (Biderman et al., 2024), and the trained models continue their own adapters through the later stage. With rank-128 adapters the drilled model fits its texts as closely and loses 1.08 and 1.14 times as much as at rank 32, but full fine-tuning is untested, and after it, or with other data and rates, relearning might not be as cheap. The base model is near its ceiling on our test problems, so reasoning training adds little accuracy here; the synthetic skill shows that the picture changes when the procedure is new, but it is a single task defined by a token budget. Training on gpt-oss-120b's traces moved the model only slightly toward it (its held-out negative log-likelihood fell 1.2% below the base's). Our realistic stage is one epoch of one dataset at one learning rate, and replay was tested in one gentle stage. At 35B and on Nemotron, both mixtures of experts, the stress tests also cost the base and the once-trained model accuracy (Table 3). Finally, we ran the 9B design three times: the direction held in every run, but under the harsh stage the size of the break varied almost fivefold between drilled sets.

AI use statement

We used an AI system (Claude, Anthropic) as a tool throughout this project. It wrote the code under our supervision, to a structure we specified down to the details. The research decisions are ours: the question, how to approach it, and how to deal with the weaknesses and problems that came up along the way. We often talked these through with the AI at length before deciding. The AI also wrote the text of the paper under our direction at every step. We decided how each section is structured, how each result and figure presents the data and how each point is worded, and we went through the draft paragraph by paragraph and line by line until every sentence said what we meant. In the code and in the paper alike, the AI served to implement and express our work, and we take full responsibility for all of it.

Ethics statement

The study uses public datasets under their licences: MATH-500 (MIT), NuminaMath-TIR through the Tülu 3 SFT mixture (Apache 2.0; the mixture is ODC-BY), AIME 2025 and 2026 through MathArena (CC BY-NC-SA 4.0), No Robots (CC BY-NC 4.0), and, for the broad chat mix, other sources of the Tülu 3 SFT mixture, each under its own licence. Third-party data are released as identifiers only, with code that rebuilds them from the public sources; the arithmetic tasks are synthetic. The study involves no human subjects.

Reproducibility statement

Every experiment was registered before its data: the records give the design, the predictions and their thresholds, and Appendix O lists every prediction with its outcome. Analyses not registered in advance are labelled post hoc in the text or listed as such in Appendix O. The supplementary material holds an anonymized repository with these records, the code, the frozen outputs and the number registry. Every number in the paper is written by code into a registry that names its source: the frozen outputs for results, and the records for design values, registered thresholds and the few results that only a record states. An independent audit recomputed the main results from the raw run files. The arms and their training are in Section 2 and Appendix A, the math data pipeline in Appendix L, the main-phase arms and the Nemotron chat template in Appendix M, the arithmetic task and its scoring in Appendices H and J, the six revision experiments (one epoch, replay, relearning, sharpening, two more runs and the synthetic skill) in Appendix N, and the prompt formats and sampling settings in the repository. The runs the paper reports used about $2,437 of Tinker compute at list prices (the teacher's traces at assumed rates); `records/compute.md` in the repository has the breakdown.

References

Anonymous. An earlier replay study of skills acquired by supervised fine-tuning and by reinforcement learning, 2026. Anonymized for review; its package is included in the supplementary material.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions, 2025. URL https://arxiv.org/abs/2505.23281.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2405.09673.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. How do large language models acquire factual knowledge during pretraining? In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.11813.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.18874.

Jessica Chudnovsky, Joshua Kazdan, Noam Levi, Rylan Schaeffer, Yegor Denisov-Blanch, Bo He, Mehmet Donmez, Sanmi Koyejo, and David Donoho. Internal data repetition destroys language models, 2026. URL https://arxiv.org/abs/2606.24998.

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

Lawrence Feng, Gaurav R. Ghosal, Jacob Mitchell Springer, Ziqian Zhong, and Aditi Raghunathan. Early data exposure improves robustness to subsequent fine-tuning. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2605.12705.

Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D. Goodman. Stream of search (SoS): Learning to search in language, 2024. URL https://arxiv.org/abs/2404.03683.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STaRs, 2025. URL https://arxiv.org/abs/2503.01307.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2021. URL https://arxiv.org/abs/2103.03874.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. Scaling laws and interpretability of learning from repeated data, 2022. URL https://arxiv.org/abs/2205.10487.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Shengyuan Hu, Yiwei Fu, Zhiwei Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned LLMs via benign relearning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fMNRYBvcQN.

Audrey Huang, Adam Block, Dylan J. Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.01951.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=A0HKeKl4Nl.

Feiyang Kang, Michael Kuchnik, Karthik Padthe, Marin Vlastelica, Ruoxi Jia, Carole-Jean Wu, and Newsha Ardalani. Quagmires in SFT-RL post-training: When high SFT scores mislead and what to use instead. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.01624.

Dawid J. Kopiczko, Sagar Vaze, Tijmen Blankevoort, and Yuki M. Asano. Data repetition beats data scaling in long-CoT supervised fine-tuning, 2026. URL https://arxiv.org/abs/2602.11149. arXiv:2602.11149v2, 9 August 2026.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.10105.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training, 2024. URL https://arxiv.org/abs/2411.15124.

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 TIR. https://huggingface.co/AI-MO/NuminaMath-TIR, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050. Source of the 500-problem MATH test subset (MATH-500).

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Ngoc-Hieu Nguyen, Parshin Shojaee, Phuc Minh Nguyen, Nan Zhang, Chandan K. Reddy, Khoa D. Doan, and Rui Zhang. Why do reasoning models lose coverage? The role of data and forks in the road. arXiv:2605.17026, 2026.

NVIDIA. Nemotron 3 Nano: Open, efficient mixture-of-experts hybrid Mamba-Transformer model for agentic reasoning. arXiv:2512.20848, 2025. Model: NVIDIA-Nemotron-3-Nano-30B-A3B-BF16, https://huggingface.co/nvidia/NVIDIA-Nemotron-3-Nano-30B-A3B-BF16.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. arXiv:2508.10925, 2025.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6Mxhg9PtDE.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026. Models: Qwen3.5-9B-Base and Qwen3.5-35B-A3B-Base, https://huggingface.co/Qwen/Qwen3.5-9B-Base.

Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. No robots. https://huggingface.co/datasets/HuggingFaceH4/no_robots, 2023.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html.

Zhiwen Ruan, Yun Chen, Yutao Hou, Peng Li, Yang Liu, and Guanhua Chen. Unveiling overmemorization in finetuning LLMs for reasoning tasks, 2025. URL https://arxiv.org/abs/2508.04117. arXiv:2508.04117v2, 28 September 2025.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6107–6122, 2022. URL https://aclanthology.org/2022.emnlp-main.410/.

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

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.04259.

Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. Overtrained language models are harder to fine-tune. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56719–56789. PMLR, 2025. URL https://arxiv.org/abs/2503.19206.

Thinking Machines Lab. Announcing Tinker. https://thinkingmachines.ai/news/announcing-tinker/, 2025. 1 October 2025.

Lukas Twist, Helen Yannakoudakis, and Jie M. Zhang. Reasoning-trace collapse: Evaluating the loss of explicit reasoning during fine-tuning, 2026. URL https://arxiv.org/abs/2605.21127.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Base models know how to reason, thinking models learn when. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.07364.

Fufang Wen and Shichang Zhang. Retention analysis of edited knowledge after fine-tuning. arXiv:2507.14198, 2025.

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. Self-distillation bridges distribution gap in language model fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://arxiv.org/abs/2402.13669.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is more for reasoning. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=T2TZ0RY4Zk.

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

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2501.13453.

A Arms and training

The appendix uses short names for the models. D, O and P are the drilled, once-trained and fresh-solution models, and a suffix gives the number of updates, so D-u140 is the drilled model after 140 updates. D-T, O-T and P-T are the same three models trained on gpt-oss-120b's traces, D2 drills a second set of problems, and D-r128 uses rank-128 adapters. O-sharp is the sharpened model of Section 5, and D-S, O-S and P-S are the three models of the synthetic skill (Appendix N). On the arithmetic task, M0 is the base model with a format adapter, T the teacher trained from it by RL, S the teacher's clone, and U and F the models drilled on the teacher's unfiltered and correct samples. U2 drills a second set of the teacher's samples, FA025, FA1 and FN are the exploratory anchor arms of Appendix G, and R-P and R-S are the two arms of the earlier replay study described in Appendix H; a suffix such as -u140 or @20 gives the number of updates.

Table 2 lists every arm. The math arms train on the base model's own correct forced solutions, drawn at temperature 1 and top-p 1 and kept if correct and uncapped (Appendix L); D uses the first 579 kept solutions, a random subset of O's 4,480. On the arithmetic task, S learns from 6,400 samples of T, 8 per prompt, kept whether right or wrong. Two arms from an earlier replay study (Anonymous, 2026), trained on an RL run's own traces and on solver derivations, are described in Appendix H.

B Replications in detail

Table 3 extends Table 1 to every arm, and Table 4 gives every state of the 35B and Nemotron runs, including the drilled model at 1.1 and 3.3 passes, and AIME.

Table 2: Acquisition arms. Passes: how many times the average training text was seen. All arms use learning rate 10^{-4}, Adam without clipping and a token loss normalized by the set's mean length, in batches of 32; FA and FN add a second stream of 32 examples per batch. The 35B and Nemotron runs repeat the base, D and O on their own solutions.

Task	Arm	Trained on	Updates	Passes
Math	D	579 of the base model's correct solutions	20–140	1.1–7.7
	O	4,480 of the base model's correct solutions	60, 140	0.43, 1.0
	P	D's 579 problems, a fresh correct solution at each visit	140	1.0
	D2	579 other solutions from O's set	140	7.7
	D-r128	D's solutions, rank-128 adapters	140	7.7
	D-T	gpt-oss-120b traces for 575 problems	20, 140	1.1, 7.8
	O-T	gpt-oss-120b traces for 4,441 problems	140	1.0
	P-T	D-T's problems, a fresh trace at each visit	140	1.0
	O-sharp	O's problems, the base model's solutions at temperature 0.5	140	1.0
	D, O, P, two more runs	as D, O and P, with a new subset, orders and adapter seed	140	7.7, 1.0, 1.0
Base-7 skill	D-S	579 problems, one worked solution each	140	7.7
	O-S	4,480 problems, one worked solution each	140	1.0
	P-S	D-S's problems, a different solution at each visit	140	1.0
Arithmetic	T	on-policy RL	30	–
	S	6,400 samples of T, unfiltered	230	1.15
	U	579 samples of T, unfiltered	20–140	1.1–7.7
	F	579 correct samples of T	20–140	1.1–7.7
	FA1, FA025	F's samples plus 32 samples of M0 per batch	140	7.7
	FN	F's samples plus 32 No Robots examples per batch	140	7.7
	U2	579 other samples of T	140	7.7

The later stages at 35B. There instruction tuning costs every model more than on the 9B: the once-trained model goes from 96.7% to 91.0%, and the untrained base, which starts from a fresh adapter, from 96.0% to 48.1%.

Where the break sets in. On the 9B it comes between 3.3 and 7.7 passes: at 3.3 passes the drilled model loses 1.1 points [−0.9, 2.9] under instruction tuning and at most 2.4 [0.2, 4.5] under the answer-only stage, although its B has risen to 0.016, 3.4% of the base model's negative log-likelihood. On the other two models it comes earlier. At 3.3 passes the 35B's drilled model loses 16.6 points [13.5,

Table 3: Every arm at handoff and after twenty updates of one later stage. Forced accuracy (%) within 16,384 tokens; free accuracy at handoff. B is a percentage of the starting model's own per-token negative log-likelihood (9B math: 0.459 nats; 35B: 0.499; Nemotron: 0.640; arithmetic, scored to 4,096 tokens: 0.390). Math: MATH-500 levels 3–5, four forced draws. Teacher traces: the same design on gpt-oss-120b's traces, where B is descriptive only. Arithmetic: 192 targeted problems, two draws; its B is scored on samples drawn at top-p 0.95, which gives every checkpoint a shared offset, so small negative values occur (Appendix I). Beyond Table 1, the table adds the second subset (D2), rank 128 (D-r128), the 35B and Nemotron runs, Nemotron's starting model, the arithmetic task with its replicate U2, and the exploratory anchor arms FA025, FA1 and FN (Appendix G). The last column is the share of forced responses (%) at the 16,384-token cap after the answer-only stage. P-u140 saw each problem 7.7 times with a fresh solution each time.

			At handoff			Forced after		At cap
Task	Model	Passes	Free	Forced	B (%)	Instr. tuning	Answer-only	after AO
Math, 9B	Base	–	74.7	94.7	0	95.6	91.7	7.0
	D-u20	1.1	95.0	95.4	0.1	94.5	94.3	5.8
	D-u60	3.3	97.3	95.1	3.4	94.0	92.8	4.2
	D-u140	7.7	95.2	95.4	24	59.3	22.5	2.6
	O-u140	1.0	94.1	94.6	0.1	94.7	94.8	4.5
	P-u140 (fresh)	1.0	94.8	94.6	0.1	94.9	94.2	5.0
	D2-u140	7.7	95.2	93.9	22	81.4	5.5	2.1
	D-r128-u140	7.7	94.1	95.6	26	56.8	12.9	2.5
Teacher traces	D-T-u20	1.1	95.0	96.2	4.6	95.4	94.3	3.6
	D-T-u140	7.8	93.9	95.0	37	33.7	1.1	0.9
	O-T-u140	1.0	95.7	96.3	5.3	94.6	94.8	4.6
	P-T-u140	1.0	95.5	96.0	5.8	95.4	95.4	4.0
Math, 35B	Base	–	82.6	96.0	0	48.1	93.7	4.0
	D-u140	7.7	95.2	94.9	32	50.8	21.5	26.6
	O-u140	1.0	97.1	96.7	0.2	91.0	95.4	5.1
Nemotron	Start	–	72.2	93.6	0	83.6	94.1	7.6
	D-u140	7.7	93.2	93.4	18	22.3	54.2	8.5
	O-u140	1.0	93.7	93.4	0.0	81.4	94.2	7.1
Arithmetic	M0	–	–	77.6	0	–	73.4	22.1
	T-u30 (RL)	–	–	78.1	0.0	–	74.7	18.0
	S-u230 (clone)	1.15	–	77.9	−0.9	–	75.8	16.1
	U-u140	7.7	–	75.5	25	–	6.0	93.2
	F-u140	7.7	–	81.5	36	–	11.7	87.5
	U2-u140	7.7	–	70.6	39	–	6.5	92.2
	FA025-u140	7.7	–	78.6	5.7	–	27.9	71.1
	FA1-u140	7.7	–	75.3	2.2	–	48.2	49.0
	FN-u140	7.7	–	85.7	3.4	–	25.3	72.4

19.8] under the answer-only stage, at a B of 4.3% of its base's, and Nemotron's loses 25.8 [21.9, 29.8] under instruction tuning at only 0.8%, where its once-trained model also falls from 93.4% to 81.4% and its starting model from 93.6% to 83.6%. On the arithmetic task the checkpoints at 3.3 passes, at 3.2–3.4% of M0's, lost clearly more than the 9B math model at the one budget where both tasks were read, 4,096 tokens (Appendix K).

AIME. On AIME 2025 and 2026 (59 problems, four draws, within 32,768 tokens) the fresh-solution model goes from 72.0% to 75.4% after instruction tuning and the 9B base from 75.8% to 71.6%. With 59 problems the interval is about ±13–16 points, so AIME is descriptive. The contrast repeats at 35B and on Nemotron, where the untrained 35B base also collapses (Table 4).

***B* in the replications.** The drilled model's B is 0.160 at 35B and 0.114 on Nemotron, against 0.0009 and 0.0002 for the once-trained ones, with 57% and 61% of it at kept top tokens. Across the five states of each run, B ranks the forced loss at Spearman 1.0 and 0.4 on the 9B, −0.1 and 0.7

Table 4: A second scale and a second family: Qwen3.5-35B-A3B-Base and Nemotron-3 Nano. Forced accuracy (%) within 16,384 tokens (AIME: 32,768), four draws, at handoff (H) and after twenty updates of instruction tuning (IT) or of the answer-only stage (AO). Nemotron's starting model is already post-trained. Last row: D-u140's excess loss over O-u140, in points.

	35B-A3B			Nemotron		
State	H	IT	AO	H	IT	AO
MATH-500, levels 3–5						
Base	96.0	48.1	93.7	93.6	83.6	94.1
D-u20	96.0	93.4	96.0	93.1	88.8	90.4
D-u60	96.3	91.6	79.6	95.6	69.8	92.4
D-u140	94.9	50.8	21.5	93.4	22.3	54.2
O-u140	96.7	91.0	95.4	93.4	81.4	94.2
AIME 2025 and 2026						
Base	79.7	5.9	–	72.9	67.8	–
D-u140	72.0	11.4	–	70.8	2.5	–
O-u140	78.8	77.5	–	74.6	64.4	–
Excess		38.3	72.1		59.2	40.0

Table 5: Forced math responses after twenty updates (884 per model). "Ends its turn": the response stops by opening a new user turn.

Model and stage	Correct	Crediting integers	Unboxed	Median tokens	Under 100 tokens	Ends its turn	Closes block	At cap
D-u140, answer-only	22.5	32.2	40.3	103	49	97	0.9	2.6
D-u140, instruction tuning	59.3	67.2	20.1	451	7.5	78	29	5.9
O-u140, answer-only	94.8	95.0	3.6	1,294	0.0	0.0	96	4.5
Base, answer-only	91.7	92.2	5.7	1,474	1.0	11	84	7.0

at 35B, and 0.7 and 0.9 on Nemotron, under instruction tuning and the answer-only stage; with five points each, all are descriptive.

C How the drilled models fail, and re-forcing

The 9B pilot (Table 5). The drilled model does not search without end; its turns end early. After the answer-only stage, 49% of its forced responses stop within 100 tokens, 97% end the turn from inside the reasoning block, which they close in 8 of 884 cases, and 40.3% leave no boxed answer; only 2.6% reach the token limit, and no response uses the stage's `<answer>` format. Crediting a correct final integer even without a box adds 9.7 points to its accuracy after the answer-only stage and 7.9 after instruction tuning; with both models scored that way, its excess loss over the once-trained model is still 64.0 and 29.1 points.

The arithmetic task. After the answer-only stage the drilled searches run to the 16,384-token cap (336 of F's and 358 of U's 384 responses). A scan of the capped texts for any expression the verifier accepts finds one in only 9 of F's and 7 of U's: they search and do not find.

On AIME after instruction tuning, the drilled model fails both ways: 28.8% of its responses reach the 32,768-token cap, 53% of those in a loop, and 36.4% leave no boxed answer. Within 16,384 tokens the base model's AIME handoff score is 51.7%, against 75.8% within 32,768.

The main-phase models. They fail through the same mix of early stops and loops, in proportions that vary. The second subset and the rank-128 model end their forced turns even sooner than the pilot's drilled model, with medians of 26 and 59 tokens after the answer-only stage. After it the 35B's drilled model has a median of 270 tokens and 62% of its responses end within 1,024 tokens, while 26.6% reach the cap, 98% of those in a loop, and 13.8% leave no boxed answer. Nemotron's

drilled model fails under instruction tuning with short, often unboxed answers: 50.5% unboxed, a mean of 1,138 tokens and 1.2% at the cap.

The re-forcing probe. It reuses the pilot's forced samples of the base model, D-u140 and O-u140 at handoff and after each later stage (884 per model and stage); no new first pass is drawn. A sample that ended its turn before the 16,384-token cap is continued: its end of turn is removed, "\n\nWait" is appended, and sampling resumes from the whole sequence with the evaluation's settings and the remaining budget, at most 8 times and never past 16,384 tokens in all. The last boxed integer of the final text is scored. Under the budget rule, the primary one, a sample is continued while it has fewer than 1,024 tokens or no boxed answer; under the answer rule, while it has no boxed answer. Neither rule looks at correctness, and both apply to every model. The registered prediction was that, under each stage, D-u140's excess over O-u140 with every model under the same rule stays at least half the registered excess, with its interval above zero; it holds under both stages and both rules (Table 7). Under the budget rule 74% of the drilled model's responses after instruction tuning, and 91% after the answer-only stage, were continued. After the answer-only stage its continued responses ran long without finding more answers (median 6,688 tokens, 47.4% at the cap). The answer rule, which continues only unboxed responses, recovers more there (37.6%). The cue itself costs the other models 1.1–2.5 points, and the base model 9.3 after the answer-only stage.

Table 6: Re-forcing the drilled 9B model. Top: D-u140's excess loss over O-u140, in points with 95% paired intervals, with every model scored under the same rule; the first pass is the registered result. Bottom: D-u140's forced accuracy (%). Budget rule (primary): a response that ends before 1,024 tokens or without a boxed answer is continued with "Wait". Answer rule: only unboxed responses are continued.

	Instr. tuning	Answer-only
Excess over O-u140		
First pass	36.2 [31.9, 40.4]	73.1 [69.3, 76.7]
Budget rule	21.3 [16.9, 25.7]	68.3 [64.3, 72.2]
Answer rule	26.5 [22.3, 30.5]	58.0 [53.7, 62.2]
D-u140, forced accuracy		
First pass	59.3	22.5
Budget rule	71.7	26.1
Answer rule	69.0	37.6

Table 7: Forced accuracy (%) on MATH-500 levels 3–5 under re-forcing: first pass, budget rule and answer rule.

State	First pass	Budget rule	Answer rule
Base	94.7	92.8	94.7
Base, instr. tuning	95.6	93.3	95.7
Base, answer-only	91.7	82.5	92.3
D-u140	95.4	93.7	95.4
D-u140, instr. tuning	59.3	71.7	69.0
D-u140, answer-only	22.5	26.1	37.6
O-u140	94.6	92.9	94.6
O-u140, instr. tuning	94.7	92.2	94.7
O-u140, answer-only	94.8	93.7	94.8

D The decision to reason

Free, forced and the decision. In free generation the model completes the prompt as it chooses; in forced generation we prefill the response with `\n<think>\n`, which opens a reasoning block. The math evaluations draw four forced and two free samples per problem. The decision to reason is the share of free responses that open the block. Nemotron's later stages train on replies with an empty reasoning block, so for Nemotron opening the block no longer marks reasoning, and we read its

decision from free accuracy; on the arithmetic task we read it from the share of openings that skip the search (Appendix H).

The decision goes first. Under instruction tuning no model of the 9B pilot opens a reasoning block when it answers freely, and free accuracy falls to 22.4–33.5%, yet with the block opened for it every model that was not drilled keeps its competence: the base gains 0.9 points and the once-trained model 0.1 (Appendix H, Figure 6b). At 35B no model opens the block either, though there the base and the once-trained model also lose forced accuracy. On the arithmetic task the teacher, the clone and M0 skip the search on half of their openings within about five updates of the answer-only stage, and drilling moves this clock earlier, while with the block opened they still solve 73.4–75.8% (Appendix H, Figure 6a). The decision does not always go: under the answer-only stage the 9B once-trained model keeps 74.2% free accuracy, often reasoning in plain text, and the registered prediction that free accuracy falls more than forced fails for the 35B base under instruction tuning and for Nemotron's starting model under the answer-only stage.

E The later stage's size

Table 8: Excess loss of forced accuracy, in points with 95% paired intervals: MATH-500 for the 9B model unless noted, and the arithmetic task's targeted problems. A positive excess means that the first model lost more than the second between handoff and the end of the later stage. The stress tests run 20 updates at 3×10^{-4}; one epoch is 267 updates at 10^{-4}.

Comparison	Later stage	Excess	Interval
Drilled vs. once-trained	instructions, one epoch	9.6	[6.9, 12.4]
Fresh solutions vs. once-trained	instructions, one epoch	−0.2	[−1.9, 1.6]
Drilled vs. once-trained	instructions, stress test	36.2	[31.9, 40.4]
	answer-only, stress test	73.1	[69.3, 76.7]
Drilled vs. fresh solutions	instructions, stress test	36.4	[32.4, 40.4]
	answer-only, stress test	72.5	[68.8, 76.0]
Fresh solutions vs. once-trained	instructions, stress test	−0.2	[−2.0, 1.5]
	answer-only, stress test	0.6	[−1.7, 2.8]
Sharpened vs. once-trained	instructions, stress test	−1.8	[−4.1, 0.3]
	answer-only, stress test	0.9	[−1.6, 3.4]
Teacher traces: drilled vs. once	instructions, stress test	59.6	[55.5, 63.7]
	answer-only, stress test	92.4	[89.4, 95.1]
Teacher traces: fresh vs. once	instructions, stress test	−1.0	[−2.6, 0.5]
	answer-only, stress test	−0.8	[−2.7, 1.1]
Drilled vs. once-trained	instructions, 10^{-4}, 60 updates	13.1	[9.8, 16.5]
	chat mix, 10^{-4}, 60 updates	10.3	[7.5, 13.1]
	instructions, 3×10^{-4}, 60 updates	55.2	[50.9, 59.4]
Teacher traces: drilled vs. once	instructions, 10^{-4}, 60 updates	6.9	[4.0, 10.0]
	chat mix, 10^{-4}, 60 updates	9.3	[6.1, 12.2]
Drilled vs. once-trained, replay	instructions, 10^{-4}, 60 updates	1.1	[−0.9, 3.3]
Teacher traces, replay	instructions, 10^{-4}, 60 updates	−0.9	[−2.8, 0.8]
Second run: drilled vs. once	instructions, stress test	61.1	[57.0, 65.2]
Third run: drilled vs. once	instructions, stress test	48.9	[45.2, 52.5]
Second drilled set vs. once	instructions, stress test	12.6	[8.8, 16.3]
	answer-only, stress test	88.6	[85.5, 91.5]
Rank 128: drilled vs. once	instructions, stress test	38.9	[35.0, 42.9]
	answer-only, stress test	82.9	[79.6, 86.1]
35B-A3B: drilled vs. once	instructions, stress test	38.3	[33.7, 43.0]
	answer-only, stress test	72.1	[68.0, 75.9]
Nemotron: drilled vs. once	instructions, stress test	59.2	[54.5, 63.8]
	answer-only, stress test	40.0	[35.6, 44.6]
Arithmetic: F vs. the clone	answer-only stage	67.7	[60.2, 75.3]
Arithmetic: U vs. the clone	answer-only stage	67.4	[60.9, 74.0]

The stages. Each longer stage runs 60 updates of 32 with a fresh optimizer and sees each example once. On the instructions it uses 1,920 distinct No Robots examples, the default stage's 640 first and in the same order, so its first 20 updates at 3×10^{-4} replay the default stage. At 10^{-4}, 60 updates match the total step, learning rate times updates, of 20 at 3×10^{-4}, a first-order match under Adam. The broad chat mix has 1,920 distinct single-turn rows of the Tülu 3 SFT mixture (Lambert et al., 2024) from its non-math sources, rendered like the instructions: Aya 361, FLAN 356, Evol-CodeAlpaca 344, WildJailbreak 203, WildGuardMix 182, WildChat 164, PersonaHub code 119 and instruction following 98, CoCoNot 35, SciRIFF 28, OASST 21 and TableGPT 9. A broad pattern for mathematics (a boxed answer, a fraction, inline mathematics, "solve" or "calculate") matches 200 rows (10.4%), mostly code tasks that calculate something and grade-school word problems; worked math word problems make up about 1–1.5% of the mix. The sources are non-math, but the mix is not free of mathematics.

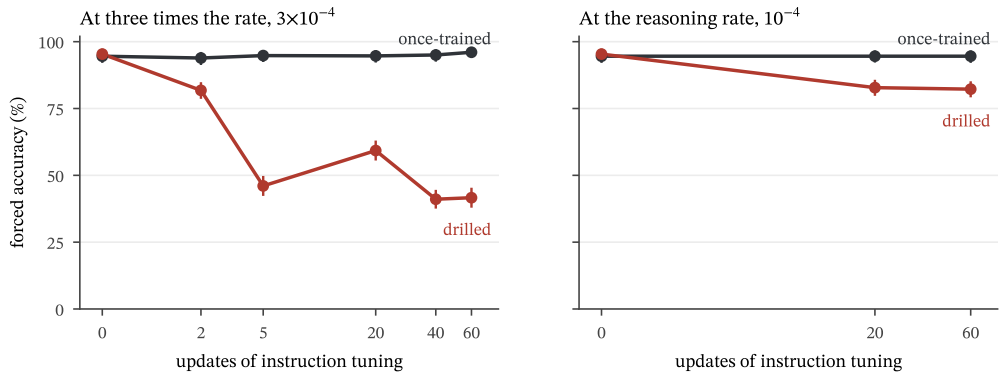

Figure 4: Forced accuracy (%, 95% intervals) of the drilled and once-trained 9B models through 60 updates of instruction tuning, at 3×10^{-4} and at the reasoning rate, 10^{-4}, on 1,920 distinct examples whose first 640 are the default stage's. At 3×10^{-4} the states after 2 and 5 updates come from a replay of the default stage whose training losses match it within 0.1%, and those after 40 and 60 from a run that replays its first 20 updates exactly, so the points trace one trajectory; lines join the measured points.

Accuracy (Figure 4). At the default rate the drilled model's forced accuracy is 59.3% after 20 updates, 41.1% after 40 and 41.6% after 60, while the once-trained model holds (94.7–96.0%). At 10^{-4} the drilled model is at 82.8% and 82.2% after 20 and 60 updates, and the once-trained model at 94.6% and 94.6%. With the teacher's traces at 10^{-4}, D-T is at 89.7% and 86.8% and O-T at 96.6% and 94.9%. After 60 updates on the broad mix, D and D-T are at 85.4% and 85.3%, O and O-T at 94.9% and 95.8%. The registered predictions that both broad-mix excesses have intervals above zero, and that O and O-T lose at most 5 points, hold; the secondary check that each excess reaches 10 points holds for D (10.3) and fails for D-T (9.3).

Training loss. Over the first five updates of the broad mix the mean loss per example is 1.30 for D and 1.30 for D-T, against 1.18 for O and 1.24 for O-T; over the last five it is 0.97, 0.97, 0.96 and 0.96. On the instructions at 10^{-4}, D-T goes from 1.85 to 1.74 and O-T from 1.80 to 1.74.

Failure form (update 60, descriptive). After 60 updates at 10^{-4}, on the broad mix and, for the teacher traces, on the instructions, the drilled models write shorter responses than the once-trained ones (median 597–749 tokens against 1,166–1,407) and close the reasoning block in only 29–38% of responses, against about 96%: the answer usually stays inside the block and the turn ends there. Their answers are unboxed in 4.4–7.0% of responses, against 2.4–3.3%. On the broad mix most of the gap is wrong answers, not missing ones; for the teacher traces on the instructions about half of it is missing answers. In every state 2.3–4.5% of responses reach the cap.

F What the later stage does to the drilled model

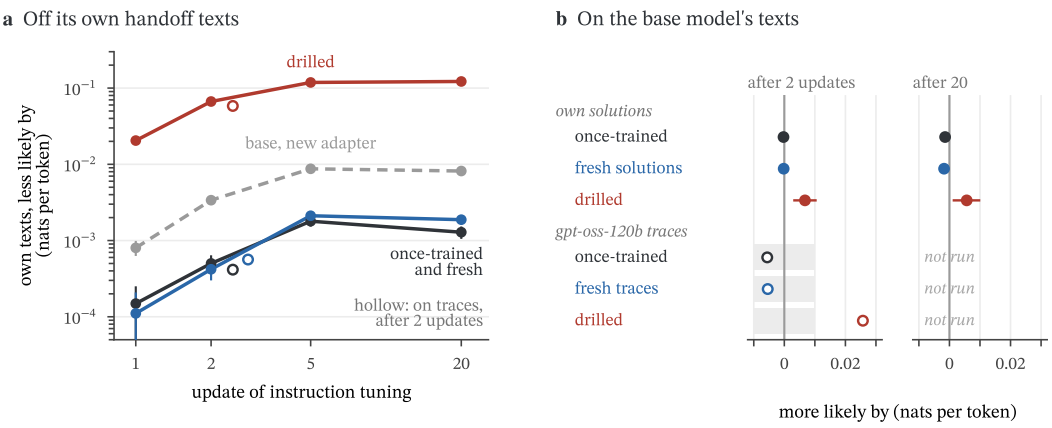

Figure 5: How far instruction tuning at 3×10^{-4} moves each 9B model. (a) How much less likely each model's own handoff texts become (the KL from its handoff, nats per token, log scale, 95% intervals) after 1, 2, 5 and 20 updates; the base, which starts from a fresh adapter, dashed; hollow: the models trained on gpt-oss-120b's traces, after 2 updates. (b) How much more likely the base model's texts become under each model after 2 and 20 updates; the shaded band is the one registered for the models trained on traces, and the drilled one falls outside it.

The alternatives at handoff. The arithmetic task shows what happens to the alternatives. Where M0 sampled one of its less likely tokens, the drilled checkpoints keep only 0.25–0.41 of the probability M0 gave it, while the teacher and the clone keep 0.91–0.98 (Figure 7b).

After handoff. Under instruction tuning every model fits the new task about equally well from the second update on, with training losses at updates 2–5 within 4.0% of one another. Yet the stage drives only the drilled model off its own solutions (Figure 5a). After two updates the texts it wrote at handoff are 0.067 nats per token less likely under it, against 0.0005 and 0.0004 for the once-trained and fresh-solution models.

From update 5 on, the drilled model gives its own solutions a lower likelihood than the untrained base does. After 20 updates their negative log-likelihood is 0.403 nats per token under the drilled model and 0.357 under the base. Part of this drop is expected, because any update that spreads probability lowers a peaked model's likelihood of its own samples. If the stage simply undid the drilling, however, those solutions would become no less likely than they are under the base. On the broad chat mix, by contrast, the drilled models start with a higher training loss than the once-trained ones and catch up only over the last five updates (Appendix E).

We had also predicted that the stage would move the drilled model further than the others on the base model's texts, and that prediction failed (Figure 5b). The base model's reasoning instead becomes slightly more likely under the drilled model, by about 0.006 of the 0.111 nats per token it had lost. Its accuracy still falls, and it does not come back within 60 updates (Appendix E, Figure 4).

An out-of-sample test. We then registered this pattern as a prediction for the models trained on a stronger model's traces. Its first part held. After two updates the drilled model's own solutions lose 0.058 nats per token, 141 times the once-trained model's 0.0004.

Its second part failed. We had predicted that the drilled model's likelihood of the base model's reasoning would change by less than 0.01 nats per token after two updates. It rose by 0.026, while under the once-trained and fresh-trace models it fell by 0.0056 and 0.0054. So the large drop on the model's own solutions replicated, and so did the movement toward the base, which we had not predicted in either setting. In both settings the drilled model moved toward the base's reasoning without regaining the base's competence.

One post hoc explanation, which we have not tested, is that drilling narrows the model's reasoning onto a few sharpened paths and prunes the base model's alternatives (the drilled 9B model keeps 0.48 of the probability the base gave them), and that the later stage then flattens those paths without giving back what was pruned. Two later experiments qualify it: a model sharpened three-quarters as much without repetition was not fragile (Section 5), and a few updates of reasoning training restore the drilled model (Section 4), so whatever the later stage takes is cheap to give back.

G The anchor, an exploratory test

Anchor and dilution. Four arithmetic arms keep F's drilling and change what goes with it. FA1 and FA025 add 32 fresh forced samples of M0 to every batch, weighted 1 or 0.25 in the loss; fitting M0's samples follows the sampled gradient of the divergence B (Section 2), so the anchor should pull the state toward M0 without removing any repetition. FN adds 32 fresh No Robots examples instead, which dilutes the drilling equally without that pull; it is not neutral, since instruction data alone wear down drilled competence (Appendix K). U2 repeats U on a second subset. Because the anchor rehearses M0's forced reasoning, a protected FA1 could be keeping rehearsed M0 competence rather than drilled competence. The two weights bound this reading only partly, and FN, meant to separate dilution from the anchor's pull, cannot be read, because most of its drilling did not take hold (below).

Adding fresh samples of M0's forced reasoning to every batch of F's drilling kept more of the search. After the answer-only stage, FA1 and FA025 solve 48.2% and 27.9%, against 11.7% for F-u140, differences of +36.5 points [30.5, 42.2] and +16.1 [10.9, 21.4], while their B falls from 0.139 to 0.022 and 0.0085; the registered dose order holds for both accuracy and B. The drilling still happened: the anchored checkpoints' negative log-likelihood on F's drilled texts fell by 85% and 99% as much as F-u140's. The replicate U2, drilled on a second subset, collapses as U did, to 6.5% with $B = 0.154$.

The registered test of protection is nevertheless not interpretable. Its parity check at handoff, registered within 4,096 tokens, failed for the anchored checkpoints, whose searches run longer: from 4,096 to 16,384 tokens FA1 gains 24.2 points and F 21.1, in the same probes. Within 16,384 tokens FA025 is level with F-u140 at handoff, 78.6% against 81.5%, but FA1 still trails at 75.3%. The dilution control FN, which adds instruction data instead, solves 25.3%, close to FA025, but cannot be read, because the added data kept most of the drilling from happening (36% of F-u140's fall in negative log-likelihood).

Anchoring also shows that B is not sufficient. FA1's B is 0.0085 and it keeps 0.95 of M0's probability on the alternatives M0 sampled, like a checkpoint that lasts. Yet it loses 27.1 points within 16,384 tokens, where M0, the teacher and the clone lose 2.1–4.2. The anchor trains on samples of the very distribution B is measured on, so its low B is partly by construction, and some of what drilling does lies outside what B measures.

The registered checks at both budgets (Table 9). The registered checks are: drilling happened (a fall in negative log-likelihood on F's drilled texts of at least 80% of F-u140's: FA1 85%, FA025 99%, FN 36%); forced parity at handoff within 4,096 tokens (within 5 points of F-u140's 61.7%: FA1 and FA025 fail by 10.7 and 8.1 points); and B moved (both anchored arms pass). The anchor texts are long searches, 55% of them reaching the 4,096-token cap, and the anchored checkpoints' searches run longer: within 16,384 tokens the handoff gaps shrink to 6.3 and 2.9 points, so FA025 would pass the margin and FA1 would not. With a check failed, the registered analysis reads the protection prediction as not interpretable; the dose order holds for accuracy and B, U2 replicates the collapse, and FN is not readable.

H The arithmetic task in detail

The task and its models. On a Countdown variant with an exact verifier (Gandhi et al., 2024), an RL teacher T is trained from M0, the base model with a format adapter, by group-relative policy gradients (Shao et al., 2024). A clone S imitates 6,400 of its samples, each seen 1.15 times on average, and U and F cycle 579 of its samples, unfiltered or correct only, to 7.7 passes (Appendix J gives the scoring rule). The question of this paper first arose on this task. A model trained by replaying 579 successful traces from an RL run for 7.7 passes lost nearly all of its accuracy within twenty updates of

Table 9: The anchor arms, forced accuracy (%) at both budgets. B over the full samples. Differences from F-u140 after the answer-only stage within 16,384 tokens, with paired intervals: FA1 +36.5 [30.5, 42.2], FA025 +16.1 [10.9, 21.4], FN +13.5 [7.8, 19.3], U2 −5.2 [−9.6, −1.0].

	At handoff			After twenty answer-only updates				
Arm	Forced, 4,096	Forced, 16,384	Free, 4,096	4,096	16,384	At cap	Strict	B
F-u140	61.7	81.5	55.7	4.2	11.7	87.5	4.9	0.139
FA1	51.0	75.3	62.2	25.8	48.2	49.0	13.0	0.0085
FA025	53.6	78.6	61.7	7.6	27.9	71.1	10.2	0.022
FN	58.3	85.7	55.2	4.9	25.3	72.4	6.5	0.013
U2	45.3	70.6	22.4	0.3	6.5	92.2	2.1	0.154

an unrelated later stage (R-P below), while a model trained by RL and a clone trained on thousands of its samples, each seen about once, kept their accuracy. Because the replayed model had seen each of its traces many times and the clone each sample about once, we tested the effect of repetition directly, first on this task and then on mathematics.

Two arms from an earlier replay study. The earlier study (Anonymous, 2026) trained two further checkpoints on the same 579 prompts in the same order, and we reuse its second realization. R-P copies, for each prompt, one verifier-correct and uncapped trace that an RL run wrote during its own training; R-S copies a deterministic solver derivation. Both use completion-only cross-entropy averaged per example, so R-P differs from U and F in its source, in success filtering and in loss normalization, and is compared with them by passes only. R-P follows the drilled pattern: forced, R-P-u140 goes from 61.2% to 2.6% within 4,096 tokens over twenty updates of the answer-only stage. R-S does not, and it is a counterexample to B as a warning. At 11.1 passes, R-S-u200 is comparably far from M0 ($B = 0.068$ over the full samples, against 0.097 and 0.139 for U and F), yet it loses 18.2 points in the same twenty updates, from 58.9% to 40.6%, and keeps writing derivations. Its text is short and unlike M0's, it teaches a procedure M0 lacks, and its loss is a per-example mean; which of these matters is untested.

The long horizon. We continued the answer-only stage to 480 updates from the arithmetic checkpoints, one run per origin and within 4,096 tokens. The registered predictions that forced accuracy would hold through update 80, and that the trained checkpoints would keep their margin over M0, failed (Table 28, C-1 to C-3 and C2); the records in the supplementary material give these probes. What repeats across runs is the decision clock and the drilled collapse within twenty updates, which the main text claims; it does not claim the late half-lives.

The decision. Four 8-token openings are sampled per problem, and an opening skips reasoning if it begins with the answer tag; the half-skip update is the update, interpolated between probes, at which half of them skip. Under the answer-only stage the teacher, the clone and M0 reach it after 4.1, 5.0 and 4.9 updates, and a second run of the first six updates gave 4.04 and 4.81 for the teacher and the clone; U, drilled 7.7 passes, reaches it after 2.7, and F a little later (Figure 6a). Instruction tuning flips the decision in another way: no checkpoint learns to answer first, but all stop searching and answer with one-line derivations (Appendix K).

The drilled search. After twenty updates of the answer-only stage 93.2% and 87.5% of U's and F's forced searches reach the 16,384-token cap, against 16.1% for the clone, and a four-times larger budget recovers little (Table 16). The clone learned from 6,400 samples over 230 updates and saw each prompt 9.2 times over eight distinct solutions, so the drilled checkpoints differ from it in repetition, in the prompts they cover and in completions per prompt.

The divergence (Figure 7). Scored over the full samples, the drilled checkpoints make M0's reasoning less likely by $B = 0.097$, 0.139 and 0.154 nats per token, against M0's own 0.390; for the teacher and the clone B is −0.0001 and −0.0034, slightly negative because of the top-p offset. B rises with passes, from −0.0027 to 0.012 and 0.097 for U and from −0.0024 to 0.013 and 0.139 for F. About two-thirds of the drilled checkpoints' B (68–71%) comes from positions where they keep M0's most likely token, and they change the top choice at 7.6–8.2% of positions against 1.3–2.2%.

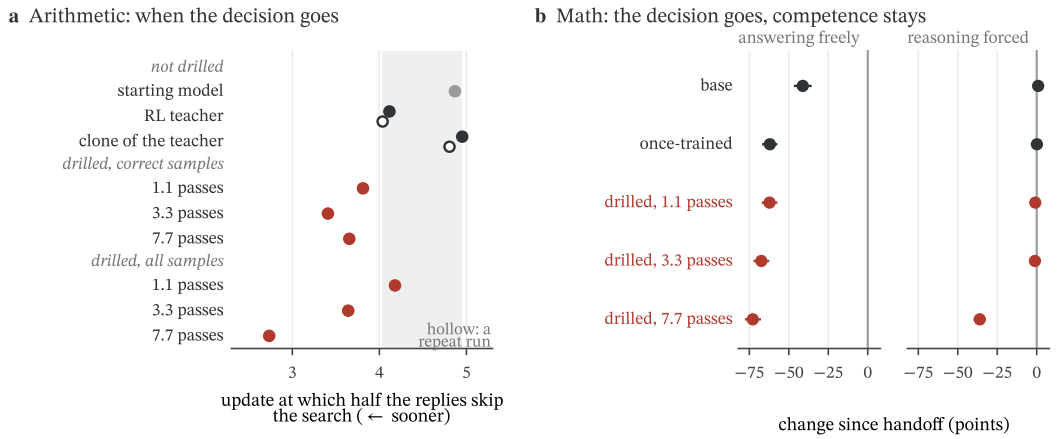

Figure 6: Later training takes the decision first. (a) The arithmetic task: the update of the answer-only stage at which half of each checkpoint's openings answer at once, without a search; hollow: a second run of the first six updates; the band spans the checkpoints that were not drilled. (b) Mathematics on the 9B: the change in free and forced accuracy between handoff and twenty updates of instruction tuning, with 95% paired intervals. Every model stops choosing to reason, and forcing gives back the accuracy of every model but the one drilled for 7.7 passes.

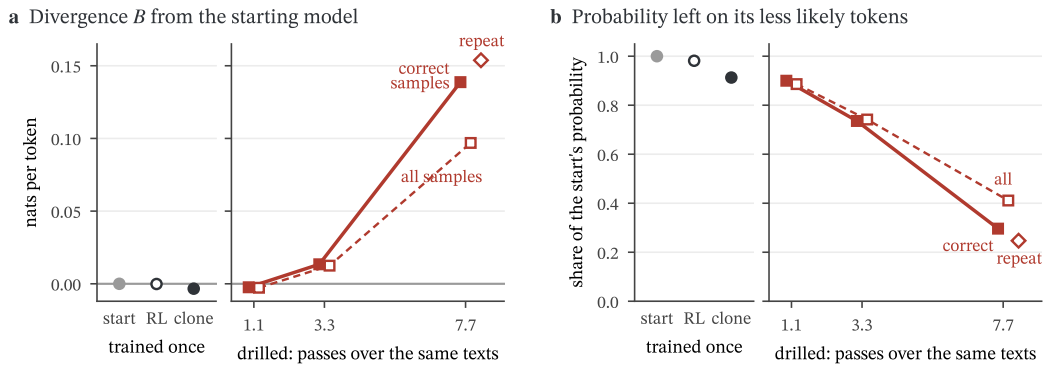

Figure 7: The arithmetic task at handoff, on M0's samples drawn at top-p 0.95 and scored to 4,096 tokens; the top-p offset (Appendix I) puts the lasting checkpoints slightly below zero in (a). Left of each panel: M0, the RL teacher and its clone, each trained about once on its texts. Right: the drilled checkpoints against passes, on the teacher's correct samples (F, filled) and on all its samples (U, open, set slightly apart), and a repeat of U (diamond). (a) B; the intervals of its differences from the clone are narrower than the markers. (b) Where M0 sampled a less likely token, the share of its probability that the checkpoint keeps.

Every checkpoint drilled to 7.7 passes without added data exceeds every lasting checkpoint's B by at least 0.046 nats per token, and the checkpoints at 3.3 passes fall between.

I Where the divergence sits, and the loss jump

The top-p offset. The arithmetic common set was sampled at top-p 0.95. On such samples B equals the divergence from M0's truncated distribution minus an offset between 0.0033 and 0.051 nats per token, shared by every checkpoint; it keeps their order and gaps but can push B below zero. Over the first 1,024 tokens the common set covers 37% of M0's tokens.

Table 10: B (nats per token) by position on M0's forced texts, first 1,024 tokens. Positions 1–3 are the forced prefix.

State	4–7	8–15	16–31	32–63	64–127	128–255	256–511	512–1,024
T-u30	0.0057	0.0003	−0.0010	−0.0017	−0.0004	−0.0006	−0.0003	−0.0002
S-u230	−0.018	−0.0040	−0.0033	−0.0044	−0.0044	−0.0039	−0.0033	−0.0032
U-u20	0.0059	−0.0026	−0.0019	−0.0030	−0.0037	−0.0029	−0.0026	−0.0027
F-u20	0.0017	−0.0021	−0.0003	−0.0025	−0.0032	−0.0030	−0.0026	−0.0024
U-u60	0.025	−0.0020	−0.0008	0.019	0.020	0.019	0.017	0.013
F-u60	0.015	0.0004	0.0072	0.0092	0.014	0.020	0.016	0.016
U-u140	0.023	−0.0014	0.0003	0.080	0.114	0.132	0.123	0.099
F-u140	0.034	0.014	0.032	0.103	0.161	0.207	0.181	0.150
R-P-u140	0.030	0.018	0.0062	0.081	0.093	0.086	0.067	0.048
R-S-u200	0.185	0.075	0.043	0.108	0.145	0.137	0.093	0.075

The position profile (Table 10). From position 8 on the lasting checkpoints stay within about 0.004 nats per token of M0. The drilled checkpoints differ only slightly over positions 4–31, where the text mostly restates the problem, then depart in the search itself, most over positions 128–255. The solver differs from the first positions on, consistent with a different kind of text. The profile is an observation; no experiment varies it. Over the same first 1,024 tokens, and over all of M0's branch points, the drilled checkpoints raise the probability of M0's top choice from 0.55 to 0.66–0.69, while the checkpoints that last raise it only to 0.58.

***B* over the full samples** (Table 11). The paper's B on the arithmetic task is scored over each common-set sample's full length, at most 4,096 tokens. Every state with positive B loses 8.2–27% of it relative to the first 1,024 tokens, and the order of the states in the dose comparison is unchanged.

Table 11: B (nats per token) scored over the full samples and over the first 1,024 tokens, with A and the share of M0's probability kept on the alternatives it sampled, both over the full samples.

State	Passes	B, full	B, 1,024	A (%)	Kept
T-u30	–	−0.0001	−0.0004	1.3	0.98
S-u230	1.15	−0.0034	−0.0035	2.2	0.91
U-u20	1.1	−0.0027	−0.0027	1.9	0.89
F-u20	1.1	−0.0024	−0.0025	1.9	0.90
U-u60	3.3	0.012	0.015	3.7	0.74
F-u60	3.3	0.013	0.016	3.8	0.74
U-u140	7.7	0.097	0.107	7.6	0.41
F-u140	7.7	0.139	0.161	8.1	0.30
U2-u140	7.7	0.154	0.172	8.2	0.25
R-P-u140	7.7	0.047	0.061	6.8	0.68
R-S-u200	11.1	0.068	0.093	7.8	0.67
FA025-u140	7.7	0.022	0.024	4.9	0.87
FA1-u140	7.7	0.0085	0.0093	3.1	0.95
FN-u140	7.7	0.013	0.015	4.1	0.78

***B* as a predictor.** Across M0 and the nine search-trained states with forced data after the answer-only stage, the rank correlation between B and the forced points lost by update 20, within 4,096 tokens, is 0.93 (0.89 with the solver). The order of these states by B is the same over the full samples and over the first 1,024 tokens, so the registered values stand. The states form three dose groups: the group label alone gives 0.92, and the number of passes 0.78. Within groups B does not rank the loss (0.4 in the low group without M0, 0.0 in the high group), and handoff accuracy does not rank it either (forced 0.23, free 0.05).

The loss jump (Table 12). One update into the answer-only stage, the drilled checkpoints' loss on the next batch more than doubles while every other origin falls. Across eleven origins the update-2 loss correlates with the change in forced accuracy by update 20 at Pearson −0.84, and across M0 and

nine search-trained states its rank correlation with the points lost is 0.94; the update-1 loss, taken before any update on the new task, gives 0.26, so the signal needs one update and is not a handoff measure. At 10^{-4} the jump is smaller (12.38 and 15.93 against 8.37–8.41), and under instruction tuning there is none (52.56 and 52.82 against 52.05–52.38 at update 2), though the drilled competence still erodes. F-u140's forced accuracy is 61.7% at handoff, 45.1% after two updates and 3.1% after twenty, within 4,096 tokens: the jump marks the start of the damage, not the damage itself.

Table 12: Loss on the next batch of the answer-only stage (summed over 32 examples); the batches are identical across origins.

Origin	Update 1	Update 2	Update 3
M0	10.45	8.06	9.31
T-u30	10.46	7.89	9.10
S-u230	10.43	7.74	10.10
U-u20	10.89	8.05	9.08
F-u20	10.97	7.86	9.11
U-u60	10.56	8.52	9.38
F-u60	10.82	8.69	9.24
U-u140	10.35	19.50	19.44
F-u140	11.77	26.91	16.19
R-P-u140	12.90	13.12	11.66
R-S-u200	21.01	12.32	9.17

J Strict and resumed scores

The rule. On the arithmetic task a response ends at its first answer tag that contains a number, and that tag is scored. A tag holding only the placeholder `<answer>EXPRESSION</answer>`, copied from the instructions, is the model restating the instructions rather than answering, so generation continues past it, within the same token budget. The strict rule ends a response at any answer tag and counts a restated placeholder as a wrong answer. The two scores differ only on responses that restate the placeholder before answering, and how often a model does that is a habit of format that the later stages change: after the answer-only stage 81% of the clone's responses restate it within 16,384 tokens, against 59% at handoff, while instruction tuning lowers M0's rate from 52% to 35% within 4,096 tokens. Strict scores therefore move with the habit as well as with competence: under instruction tuning M0's strict score rises from 13.8% to 26.3% while its resumed score stays level, and after the answer-only stage the clone's strict score falls from 26.3% to 10.9% while its resumed score holds. The rule was adopted during the decision-competence experiment after only the original samplers' long-horizon probes had been read, and every later probe records both scores.

What survives the strict rule (Tables 13, 14 and 15). The drilled checkpoint on correct solutions collapses under both rules, from 68.5% to 4.9% strict within 16,384 tokens. Strict handoff accuracy points the wrong way: within 4,096 tokens F-u140 scores 52.9% against 10.4–22.9% for the checkpoints that last. The contrast between lasting and drilled is weaker under the strict rule within 16,384 tokens: the clone goes from 26.3% to 10.9% and U-u140 from 17.4% to 0.8%, both with high restatement rates.

Held-out families at handoff, within 4,096 tokens, resumed / strict: M0 54.2 / 12.5, T-u30 61.5 / 17.4, S-u230 58.9 / 14.8, R-P-u140 60.9 / 51.6, R-S-u200 62.2 / 35.2.

K Token budgets

A sample counts as correct within a budget of N tokens only if it finished correct inside N tokens, so a long probe's first N tokens are distributed exactly as a probe capped at N. The original arithmetic probes were capped at 4,096 tokens; the cap test and the consistency probes at 16,384; math at 16,384 (32,768 for AIME).

Table 13: Accuracy (%) at handoff within 4,096 tokens: forced, resumed and strict, the share of forced responses resumed, and free accuracy. Free accuracy on the arithmetic task was probed only at this budget.

State	Resumed	Strict	Share resumed	Free
M0	54.9	13.8	52	13.5
T-u30	62.2	22.1	48	44.0
S-u230	57.0	14.6	51	37.8
U-u20	59.9	10.4	61	43.8
F-u20	58.6	22.9	43	51.8
U-u60	64.8	14.1	62	46.1
F-u60	66.1	39.6	30	65.6
U-u140	53.6	12.8	51	33.9
F-u140	61.7	52.9	11	55.7
R-P-u140	61.2	51.8	12	43.2
R-S-u200	58.9	34.6	35	39.3

Table 14: Forced accuracy (%) after twenty updates of each later stage, within 4,096 tokens, as resumed / strict.

State	Answer-only, 3×10^{-4}	Answer-only, 10^{-4}	Instr. tuning
M0	45.1 / 9.9	–	53.6 / 26.3
T-u30	59.4 / 22.9	–	–
S-u230	57.6 / 8.6	–	–
U-u20	58.1 / 25.3	61.5 / 10.4	58.9 / 26.8
F-u20	62.5 / 22.1	60.9 / 21.6	57.8 / 37.0
U-u60	46.4 / 7.6	–	–
F-u60	47.9 / 28.4	–	–
U-u140	0.3 / 0.3	29.9 / 6.8	34.9 / 27.6
F-u140	3.1 / 2.3	37.2 / 28.1	36.7 / 31.0
R-P-u140	2.6 / 0.5	–	–
R-S-u200	40.6 / 22.7	–	–

Arithmetic. At handoff every state solves 19.5–27.6 points more within 16,384 tokens than within 4,096, and all five are within 6.0 points of one another at 16,384 (Table 16). After twenty answer-only updates the budget separates the drilled states from the rest: M0, the teacher and the clone lose 2.1–4.2 points within 16,384 tokens, the drilled states about 69.8, and a four-times larger budget recovers little. The within-4,096 readings of these probes agree with the original probes within sampling error, for example 54.2% against 57.6% for the clone after the answer-only stage.

The dose at 4,096 tokens. Forced accuracy lost over twenty answer-only updates grows with passes: −1.8, −18.5 and −53.4 points for U and +3.9, −18.2 and −58.6 for F at 1.1, 3.3 and 7.7 passes, against +0.5 for the clone, −2.9 for the teacher and −9.9 for M0. Drilled minus once-trained is −51.6 [−60.4, −42.7] for U and −62.5 [−70.1, −55.2] for F; at 10^{-4} it is −25.3 and −26.8, and under instruction tuning −17.7 and −24.2.

Instruction tuning and the lower learning rate. These arithmetic probes were read only within 4,096 tokens. Under instruction tuning no checkpoint learns to answer first, but all stop searching: the checkpoints trained about once answer freely with one-line derivations (median 119–139 tokens), and forced, M0 and those checkpoints still solve 53.6–58.9%. Under instruction tuning and at a learning rate of 10^{-4}, the checkpoints trained about once lose at most 1.0 point, while the drilled ones lose far more (Table 14).

3.3 passes on both tasks. This is the one budget at which both tasks were read at 3.3 passes. The drilled 9B math model goes from 80.2% at handoff to 80.8% after the answer-only stage and 83.3% after instruction tuning, while the arithmetic checkpoints at 3.3 passes lose 18.5 (U) and 18.2 (F) points under the answer-only stage.

Table 15: Forced accuracy (%) within 16,384 tokens, as resumed / strict (share resumed).

State	Handoff	After twenty answer-only updates
M0	77.6 / 19.3 (67)	73.4 / 18.8 (61)
T-u30	78.1 / 22.9 (63)	74.7 / 34.6 (45)
S-u230	77.9 / 26.3 (59)	75.8 / 10.9 (81)
U-u140	75.5 / 17.4 (66)	6.0 / 0.8 (35)
F-u140	81.5 / 68.5 (14)	11.7 / 4.9 (20)

Table 16: Arithmetic forced accuracy (%) within each budget, from the 16,384-token probes; "Over" and "Cap" are the shares of responses longer than 4,096 tokens and at the 16,384 cap.

	At handoff			After twenty answer-only updates				
State	4,096	8,192	16,384	4,096	8,192	16,384	Over	Cap
M0	50.0	65.1	77.6	50.8	64.8	73.4	46	22.1
T-u30	58.6	71.6	78.1	55.5	65.6	74.7	41	18.0
S-u230	54.2	68.8	77.9	54.2	65.6	75.8	41	16.1
U-u140	52.9	65.4	75.5	1.8	2.9	6.0	98	93.2
F-u140	60.4	73.4	81.5	4.2	7.3	11.7	95	87.5

Mathematics (Table 17). After the answer-only stage the drilled model solves 21.8% within 4,096 tokens and 22.5% within 16,384, and only 2.6% of its responses reach the cap: on math its loss is not a budget effect. The once-trained model stays within about one point of its handoff accuracy at every budget.

L The math data pipeline

The source is Tülu 3's decontaminated NuminaMath-TIR subset (Lambert et al., 2024; Li et al., 2024), with 64,312 rows and 58,674 distinct problems; the reference answer is the last boxed answer of the reference solution. Table 18 lists the screens. Decontamination used exact matching, 13-gram overlap, punctuation-stripped 13-gram overlap and a hand-reviewed paraphrase search; its 91 hits include an AIME 2026 problem word for word and two MATH-500 paraphrases, and a second pass of exact and 13-gram matching after removal found none. Reference answers agree with the reference solutions' own code output in 93% of a 200-row sample. A problem is skipped if the base model solves it within 256 tokens after an empty think block; in calibration 83% of such short samples still wrote working until the cap, so the screen removes problems the model solves almost at once. Each remaining problem receives one forced sample at temperature 1 and top-p 1 with a 16,384-token cap, kept if correct and uncapped; D uses the first 579 kept examples, a random subset of O's 4,480. The scorer takes the last boxed answer, handles nested and escaped braces, treats an unclosed box as no answer, strips units, degrees, currency, percent signs and thousands separators, and parses an integer; it passes 30 unit examples, recovers all 311 MATH-500 reference answers, and read the final box correctly in all 20 hand-checked base solutions. Evaluation excludes MATH-500 levels 1–2, which the base model solved in full in calibration (Table 19), and one AIME 2026 problem that is in the source.

M The main-phase arms

P, fresh solutions. P visits D's 579 problems in D's cycling order for 140 updates. Each visit uses a fresh correct forced sample of the base model (top-p 1, cap 16,384), D's own sample first, and sampling stops at 8 kept samples or 24 draws per problem. 577 of the 579 problems yielded 8 distinct correct samples; the other two, with fewer, repeat 6 of the 4,480 training slots.

D2 and rank 128. D2 takes 579 of the pilot O's solutions outside D, drawn by hash, and cycles them for 140 updates; its comparator is the pilot's O-u140. The rank-128 arm uses D's data and order.

Table 17: Math forced accuracy (%) within each budget, MATH-500 levels 3–5, with the shares of responses at the cap and without a boxed answer, and the mean response length.

Model and stage	2,048	4,096	8,192	16,384	Capped	Unboxed	Mean tokens
Base, handoff	67.2	79.6	88.3	94.7	5.1	3.8	3,340
Base, instruction tuning	71.8	83.7	91.1	95.6	2.5	1.9	2,712
Base, answer-only	67.9	78.5	86.5	91.7	7.0	5.7	3,608
O-u140, handoff	68.3	80.1	88.8	94.6	5.4	3.8	3,370
O-u140, instruction tuning	67.5	81.3	89.3	94.7	4.8	4.1	3,196
O-u140, answer-only	67.9	79.5	89.6	94.8	4.5	3.6	3,162
D-u140, handoff	71.5	81.6	89.5	95.4	4.5	3.6	3,058
D-u140, instruction tuning	51.4	55.2	58.3	59.3	5.9	20.1	2,007
D-u140, answer-only	20.4	21.8	22.5	22.5	2.6	40.3	867

Table 18: Screens of the math pool, in order of removal.

Removed	Problems
Answer not a single integer	12,493
Solution boxes several different answers	2,614
Duplicate copies disagree	107
Figures	331
Prompt over 512 tokens	28
Contamination hits	91
Guessable or not single-answer	6,563
Kept	36,447

Both ranks drilled comparably: the negative log-likelihood of D's texts falls from the base's 0.453 to 0.277 at rank 32 and 0.270 at rank 128, drops within 3.8% of each other (registered margin 20%).

The 35B noise floor. A fresh LoRA scored the 35B base's texts with a mean absolute difference of 0.0034 nats per token, against a registered 0.001, because this mixture-of-experts model is not deterministic: two independently created base samplers differ by 0.0030 on average. On the run's common set, a second base sampler gives $B = 0.0000$ $[-0.00002, 0.00002]$ and a mean absolute per-token difference of 0.0003, so every state's B, O-u140's 0.0009 included, lies above the floor.

Teacher traces. gpt-oss-120b wrote traces at high reasoning effort, temperature 1 and at most 16,384 tokens; a trace was kept if it was a clean reply whose last boxed integer matched the answer, and a problem entered D-T and O-T only if the teacher solved it within its first four draws. D-T uses 575 problems and O-T 4,441; P-T gives D-T's problems about 7.99 kept traces each. The drilled model fails the same way as the 9B pilot's, with short turns: after the answer-only stage its forced responses have a median of 22 tokens, and after instruction tuning 277, with 39.6% unboxed. On AIME, D-T falls from 73.7% to 11.0% after instruction tuning, while O-T goes from 72.9% to 78.0%. The registered rival to the break, consolidation, predicted an excess under 3 points at each stage. It failed: D-T's excess is 59.6 and 92.4 points. Its reading conditions failed too. The drilling reached 0.69 of the self-trained arm's fall in training loss, against 0.8 required, and the once-trained model's held-out negative log-likelihood fell only 1.2%, against 20% required. Those conditions gated only a null reading, so they change no outcome, but they bound the cell: the student moved little toward the teacher.

Nemotron. Nemotron-3 Nano is prompted in its own chat template, which our renderer matches token for token on 7,047 prompts. Forced reasoning prefills its own reasoning opening, direct answers an empty reasoning block, and turns end on its end-of-turn token. The later stages use the same examples as the Qwen runs, as answer-only turns. A fresh LoRA reproduces the model exactly over 1,032 positions.

Table 19: Calibration of the base model, one draw per problem and mode. Direct: an empty think block with a 256-token cap.

Source	Forced	Free	Direct	Median tokens
MATH-500 levels 1–2	1.00	1.00	0.58	850
MATH-500 level 3	0.92	0.83	0.42	1,326
MATH-500 level 4	0.88	0.71	0.17	954
MATH-500 level 5	0.62	0.58	0.08	2,928
OlympiadBench (integer)	0.46	0.42	0.00	6,144 (cap)
NuminaMath-TIR (integer)	0.73	0.67	0.19	1,410

N The revision experiments

Six experiments were registered together in one record, published before any of their data; the synthetic skill was amended once, after a small smoke test and before its main samples. Each was launched only after its code was fixed and its hashes recorded. They start from the states of the main experiments or train new ones, and the math parts measure as the paper does: forced accuracy on the 221 test problems, four draws of up to 16,384 tokens, and 95% paired bootstrap intervals over problems. An excess compares the losses of two models, each from its own handoff.

One epoch of instruction tuning (Table 20). The stage is a single pass over 8,544 No Robots training rows, chosen by the selection rules of the instruction stage (all but one of the rows that pass them) in a fixed hash order: 267 updates of 32 at 10^{-4} with a fresh optimizer, evaluated after the last. The base model starts it from a new adapter. The drilled model's excess over the once-trained model is 9.6 [6.9, 12.4] and the fresh-solution model's −0.2 [−1.9, 1.6]. Both registered predictions held: the first excess is at least 5 points with its interval above zero, and the second within 3 points of zero.

Table 20: One epoch of instruction tuning at 10^{-4}: forced accuracy (%) and each model's loss, in points with its 95% interval.

Model	Handoff	After	Loss	Interval
Base, new adapter	94.7	94.5	0.2	[−1.2, 1.6]
Drilled	95.4	86.0	9.4	[6.8, 12.1]
Once-trained	94.6	94.8	−0.2	[−1.5, 1.0]
Fresh solutions	94.6	95.0	−0.5	[−1.6, 0.7]

Replay (Table 21). The stage follows the gentler schedule of Appendix E, 60 updates at 10^{-4}, but each batch of 32 holds 30 of its instruction examples, in order, and 2 of the model's own training examples, taken in its training order and cycled, with their original loss weights. For the drilled models these are their repeated texts. The drilled model's excess over the once-trained model is 1.1 [−0.9, 3.3], against 13.1 without replay, a reduction of 91%. With gpt-oss-120b's traces it is −0.9 [−2.8, 0.8], against 6.9, an excess indistinguishable from zero. The registered reading for an excess of at most 3 points with an upper bound under 6 is that replay prevents the break, and it applies to both. The prediction, a reduction of at least 50% for the model's own solutions, held.

Table 21: Replay: forced accuracy (%) at handoff and after 60 updates of instruction tuning at 10^{-4} with 6.25% of each batch replayed from the model's own training texts.

Training texts	Model	Handoff	After
Own solutions	Drilled	95.4	94.3
	Once-trained	94.6	94.7
gpt-oss-120b traces	Drilled	95.0	95.5
	Once-trained	96.3	95.8

Relearning (Table 22). The origins are the four drilled models after the stress tests, and the once-trained model after instruction tuning as a ceiling control. The math data are the base model's own correct forced solutions to 640 problems of the training pool that no model had trained on and the test set does not contain, drawn in pool order after the pilot's screened problems with the pilot's screen and keep rule. The habit data are the base model's forced responses to the first prompts of the broad chat mix of Appendix E, kept until 640 had a closed reasoning block followed by an answer; about 1–1.5% of that mix are worked math word problems. Each origin continues its own adapter at 10^{-4} in batches of 32 with a fresh optimizer, sees each example once, and is evaluated after 1, 2, 5 and 20 updates, the habit runs after 20 only. The share recovered is $R_k = (a_k - a_0)/(a_{\text{ref}} - a_0)$, where a_k is the drilled model's accuracy after k updates and a_{ref} the matching once-trained model's after the same later stage (94.6–94.8%). Both predictions held: R_5 of the drilled model after instruction tuning is at least 0.5, and the ceiling control moves by at most 0.6 points, within the registered 2. The registered readings apply: with R_5 of at least 0.75 the drilled model's competence reads as suppressed and restored, and with a habit share of at least 0.5 the paper drops the claim that the loss is more than a lost habit.

Table 22: Relearning: forced accuracy (%) after k updates of reasoning training from each broken model, the share R_5 recovered after five updates with its 95% interval, and forced accuracy after twenty updates of the habit data, with its share R_{20}. IT: the instruction stress test; AO: the answer-only stage.

Origin	$k = 0$	1	2	5	20	R_5	Habit, $k = 20$
D + IT	59.3	84.7	90.3	93.9	94.3	0.98 [0.94, 1.01]	94.0 (0.98)
D + AO	22.5	51.0	79.5	92.8	93.3	0.97 [0.95, 1.00]	92.3 (0.97)
D-T + IT	33.7	76.4	87.1	93.1	93.3	0.98 [0.94, 1.00]	–
D-T + AO	1.1	21.6	70.5	91.5	93.0	0.96 [0.94, 0.99]	–
O + IT	94.7	95.2	94.2	94.3	94.6	–	–

Sharpening without repetition (Table 23). For each of O's 4,480 problems, in O's order, the sharpened model takes the base model's first correct and uncapped forced sample at temperature 0.5 (top-p 1), drawing up to 4 times. 4,453 problems had one; the slots of the rest are filled by cycling from the start, as for the teacher's traces. It is trained exactly as O. Before any evaluation its B had to reach 0.05 nats per token, or the model would have been sampled and trained again at temperature 0.3; it reached 0.082 [0.077, 0.087]. Its excess over the once-trained model is −1.8 [−4.1, 0.3] after instruction tuning and 0.9 [−1.6, 3.4] after the answer-only stage, so the registered reading "sharpening alone does not break it" (within 3 points after both stages) applies, and the prediction that its answer-only excess would stay under 36.5 held.

Table 23: Sharpening without repetition: B at handoff (nats per token) and forced accuracy (%) at handoff and after the two stress tests.

Model	B	Handoff	Instructions	Answer-only
Sharpened	0.082	93.1	95.0	92.4
Once-trained	0.0004	94.6	94.7	94.8
Drilled	0.111	95.4	59.3	22.5

Two more runs (Table 24). Each run draws a new subset of 579 of O's problems, disjoint from D's and D2's, a new order for D, a new order for O's 4,480 texts and a new adapter initialization; the two new subsets share 91 problems. D, O and P are trained as in the first run, with P's fresh solutions sampled for the new problems under the same rule, and each is evaluated at handoff, after the instruction stress test and after 60 updates at 10^{-4} on the instructions. The registered predictions, for each run and stage, are that the drilled model's excess over the once-trained and over the fresh-solution model has its interval above zero, secondarily that it is at least 5 points, and that the fresh-solution model's excess over the once-trained model stays within 3 points of zero. All three held in both runs and both stages, with the fresh-solution model within 1.4 points of the once-trained one.

As registered, we report each run and the pooled result with its spread: over the four drilled sets, counting D2, the drilled model's extra loss in the instruction stress test averages 39.7 points (range 12.6–61.1, standard deviation 20.7), and at the gentler rate, over the three runs, 14.7 (13.1–17.8).

Table 24: The two new runs of the 9B design: forced accuracy (%) at handoff and after each stage, and the drilled model's excess over the once-trained model, in points with its 95% interval. IT: the instruction stress test; gentle: 60 updates at 10^{-4}.

Run	Model	Handoff	IT	Gentle
Second	Drilled	94.8	32.7	80.9
	Once-trained	96.2	95.1	95.4
	Fresh solutions	95.6	94.6	94.0
	Drilled's excess		61.1 [57.0, 65.2]	13.1 [10.0, 16.3]
Third	Drilled	94.6	45.9	76.1
	Once-trained	94.9	95.1	94.2
	Fresh solutions	94.2	94.5	94.9
	Drilled's excess		48.9 [45.2, 52.5]	17.8 [14.1, 21.5]

The synthetic skill: choosing the task. Two tasks were registered, in order: multiplying two base-7 numbers of five and then six digits, and sorting 12 and then 16 words under an alphabet given in the prompt. Each has an exact checker and a solver that writes several distinct worked solutions per problem, by different methods and in different step orders; for base-7 multiplication these come from long multiplication with either number on top, conversion through base 10 and back, and collecting digit products by place. The registered rule took the first setting on which the base model's forced accuracy is at most 25%. A first check on 5 problems per setting, four draws each, found that the base model solves these tasks when it is given room: within 4,096 tokens most of its responses were cut off, and within 8,192 it solved 75.0% of the five-digit products, 55.0% of the six-digit ones and 85.0% and 60.0% of the sorts. Before any sample at a shorter budget we redefined the skill as solving within 2,048 tokens, a budget set by the worked solutions (mean 871 tokens for five digits, longest 1,937) and not by any result of the base model, and kept everything else as registered. At that budget the base model solves 0.2% of 300 five-digit calibration problems, so that setting was chosen.

The synthetic skill: training and results (Table 25). D-S trains on 579 problems with one worked solution each for 140 updates, O-S on 4,480 problems once, and P-S on D-S's problems with a different worked solution at each visit; the solutions of each problem are shuffled, so every model sees a mix of methods. A registered gate required O-S to beat the base model by at least 30 points before the other models were trained; it did so by 66.1 [63.3, 68.8], with 0.1% of responses at the budget. The registered outcomes compare D-S with O-S after each stage: a break if D-S ends at least 10 points below O-S with its interval above zero, consolidation if it ends within 3, and a costless remedy if P-S is within 3 points of the better of the two at handoff and after the stage. After the gentler stage D-S ends 24.2 points below O-S [20.4, 27.9], a break as worded, but its extra loss from handoff, which we computed only after the runs, is −4.2 [−9.3, 0.9], so we read the gap as one of learning and report both readings. As a share of each model's handoff accuracy the stage cost D-S 29.9%, O-S 23.4% and P-S 33.4% (also post hoc). Consolidation fails, and so does the costless remedy (P-S minus the better of the other two is −8.2 points at handoff and −12.1 after). After the stress test every model is below half a percent, so consolidation holds there trivially (0.3 points, [0.0, 0.8]). P-S's lead over D-S at handoff, 20.2 [16.2, 24.1], was not registered.

Registered predictions. Table 26 lists the eight confirmatory predictions of these experiments with their outcomes; all eight held.

O Registered predictions and outcomes

Every experiment was registered before its data, with its predictions and their thresholds, and the records are in the supplementary material. Tables 27 and 26 list the 44 confirmatory predictions on mathematics, with the rival reading T2 beside T1, and Table 28 the 26 predictions of the exploratory

Table 25: The synthetic skill: forced accuracy (%) on 300 held-out five-digit base-7 products within 2,048 tokens, four draws each, and each model's loss in the gentler stage (60 updates at 10^{-4}) with its 95% interval.

Model	Handoff	Gentle	Stress test	Loss, gentle
Base	0.2	–	–	–
Drilled (D-S)	37.9	26.6	0.1	11.3 [7.8, 14.8]
Once-trained (O-S)	66.3	50.8	0.4	15.5 [11.7, 19.3]
Fresh solutions (P-S)	58.1	38.7	0.4	19.4 [15.4, 23.3]

Table 26: Registered predictions of the revision experiments, confirmatory, one per row; columns, outcomes and claims as in Table 27. SD's secondary prediction, an excess of at least 5 points in each run and stage, also held; secondary predictions are listed in the records.

ID	Prediction	Threshold	Observed	Outcome	Claim
RS1	One epoch breaks D	excess ≥ 5, CI > 0	9.6	held	break
RS2	One epoch spares P	P − O within ±3	−0.2	held	cause
RP1	Replay mitigates, own solutions	reduction $\geq 50\%$	91%, prevents	held	remedy
RL1	Relearning is fast	R_5 of D + IT ≥ 0.5	0.98	held	competence
RL2	The ceiling control holds	O + IT within 2 at every k	at most 0.6	held	competence
SH1	Sharpening alone does less than half	AO excess of O-sharp < 36.5	0.9	held	signature
SD1	D loses more in two new runs	D − O and D − P, CIs > 0, each run and stage	over O: 61.1, 48.9 (IT); 13.1, 17.8 (gentle)	held	break
SD3	P behaves like O in two new runs	P − O within ±3, each run and stage	at most 1.4	held	cause

line on arithmetic, run before and alongside the math pilot. Of all 70, 50 held, 17 failed (7 of them in part) and 3 could not be read; 12 of the failures are on arithmetic. The predictions behind the headline claims held: the break in every cell (MP3, M1.3, M5.3, M2.1, N2, RS1, SD1), its cause (M3.2, RS2, SD3), the teacher traces (T1, T4), the dependence on the later stage (R2.1, R3a, R5.1–R5.3), relearning (RL1, RL2), replay (RP1) and sharpening (SH1). RF1 held as registered, but the relearning experiment, registered later, shows that the drilled model's competence is suppressed rather than lost, and the paper follows that reading. The failures that qualify the claims, M1.5, M5.5, R3b, T5(b) and R4.1, are stated where the claims are made. Registered conditions, alternative readings, descriptive items and secondary predictions are listed in the records. The paper also reports analyses that were not registered, as observations: the share of its training tokens the drilled model predicts (Section 5); the fresh-solution model's lead over the drilled one on the synthetic skill (Appendix N); the 35B base's collapse under instruction tuning, which starts from a fresh adapter (Section 3); the drop of the drilled model's own texts below the base's likelihood, its movement toward the base's reasoning and the broad mix's training losses (Appendices F and E); the synthetic skill's extra loss from handoff and its losses as shares of handoff accuracy (Appendix N); the failure forms, AIME and strict scores (Appendices C, E and J); the loss jump (Appendix I); and the anchor arms, which are exploratory (Appendix G).

Table 27: Registered predictions on mathematics (confirmatory), one per row. Threshold: the registered criterion; Observed: the result, in points of forced accuracy unless marked. Outcomes: held; failed; failed* (in part); rival, not supported (T2 was registered as the alternative to T1, so at most one could hold). Claim: break (the drilled model breaks where the once-trained one does not), cause (repeated text is the cause), teacher (teacher traces), competence (whether the competence is lost or suppressed), stage (dependence on the later stage), signature (B and over-sharpening), mechanism (what the later stage does), decision (the decision to reason), remedy (replay). "As MP1" repeats the pilot's criterion; IT is instruction tuning and AO the answer-only stage.

ID	Prediction	Threshold	Observed	Outcome	Claim
Math pilot					
MP1	Drilling over-sharpens	B(D) − B(O) ≥ 5% of the base's NLL, CI > 0	24% against 0.1%	held	signature
MP2	B rises with passes	D-u20 < D-u60 < D-u140	0.1, 3.4, 24%	held	signature
MP3	The drilled model loses more, each stage	more than D-u20 and O, CIs > 0	36.2 (IT), 73.1 (AO) over O	held	break
MP5	Free accuracy falls more than forced	base and O, both stages	all four	held	decision
Fresh solutions (P)					
M3.1	P keeps B low, at parity	B(P) ≤ B(D)/2; handoff within 5	B 0.0004 against 0.111	held	signature
M3.2	Repeated text is the cause	D − P ≥ 20, CIs > 0	36.4, 72.5	held	cause
35B					
M1.1	Drilling over-sharpens	as MP1	32% against 0.2%	held	signature
M1.2	B rises with passes	as MP2	rises	held	signature
M1.3	The drilled model loses more	as MP3	38.3, 72.1 over O	held	break
M1.5	Free accuracy falls more than forced	as MP5	not for the base under IT	failed*	decision
M1.6	At least half of B at kept top tokens	at least half	57%	held	signature
M1.7	On AIME, D loses more after IT (descriptive)	direction	D 72.0 to 11.4, O 78.8 to 77.5	held	break
Nemotron					
M5.1	Drilling over-sharpens	as MP1	18% against 0.0%	held	signature
M5.2	B rises with passes	as MP2	rises	held	signature
M5.3	The drilled model loses more	as MP3	59.2, 40.0 over O	held	break
M5.5	Free accuracy falls more than forced	as MP5	not for the start model under AO	failed*	decision
M5.6	At least half of B at kept top tokens	at least half	61%	held	signature
M5.7	On AIME, D loses more after IT (descriptive)	direction	D 70.8 to 2.5, O 74.6 to 64.4	held	break
Second subset and rank					
M2.1	A second drilled subset breaks	D2 − O ≥ 10, CIs > 0	12.6, 88.6	held	break
M2.2	It over-sharpens	as MP1	22%	held	signature
N2	Rank 128 breaks too	at least half of rank 32's loss	1.08×, 1.14×	held	break
Re-forcing					
RF1	Competence is lost	re-forced excess at least half the first, CIs > 0	21.3, 68.3	held	competence
Teacher traces					
T1	Drilled traces break	excess ≥ 10 over O-T and D-T-u20, CIs > 0	59.6, 92.4 over O-T	held	teacher
T2	Rival: consolidation	excess < 3, upper bound < 10	59.6, 92.4	rival, not supported	teacher
T4	Fresh traces fix it	P-T − O-T ≤ 5	−1.0, −0.8	held	teacher
T5	Narrowing out of sample: (a) own texts, (b) the base's texts	(a) ≥ 10× O-T's; (b) within ±0.01 nats	(a) 141×; (b) −0.026	failed*	mechanism
Robustness					
R1.1	The stage moves D most on its own texts	≥ 2× O's and P's, CI > 0	0.067 against 0.0005, 0.0004 nats	held	mechanism
R1.2	Displacement comes before the collapse	D forced at update 2 > 77.4%	81.8%	held	mechanism
R1.3	Fresh solutions move like once-trained ones	P ≤ 1.5× O at updates 1, 2, 5	0.74–1.18×	held	mechanism
R1.4	Every model fits the new task equally	loss spread ≤ 10% at updates 2–5	1.7–4.0%	held	mechanism
R2.1	The break survives a third of the rate	excess ≥ 10, CI > 0	13.1	held	stage
R3a	The loss persists to update 60	excess ≥ 20, CI > 0	55.2	held	stage
R3b	The stage moves D most on the base's texts	≥ 2× O's and P's, CI > 0	−0.0068 nats	failed	mechanism
R4.1	The teacher-trace break survives a third of the rate	excess ≥ 10, CI > 0	6.9 [4.0, 10.0]	failed	stage
R5.1	The broad mix breaks D	excess CI > 0	10.3	held	stage
R5.2	The broad mix breaks D-T	excess CI > 0	9.3	held	stage
R5.3	The broad mix is routine for O and O-T	each loses ≤ 5	−0.3, 0.5	held	stage

Table 28: Registered predictions on the arithmetic task, an exploratory line run before and alongside the math pilot; B is over the first 1,024 tokens, as registered. Columns and outcomes as in Table 27; not readable: a registered check failed, or the item was never evaluated. Claims as in Table 27, plus budget (token budgets), anchor (the anchor arms), horizon (long-horizon probes) and earlier study (claims of the earlier replay study, re-examined; its records are in the supplementary material). Accuracy is forced, in percent; U and F are drilled to 7.7 passes unless marked.

ID	Prediction	Threshold	Observed	Outcome	Claim
Depth profile and cap test					
DP1	Repetition, not training amount, moves the reasoning	B(S) below B(U) and B(F), CIs < 0	-0.0035 against 0.107, 0.161	held	signature
DP2	Every lasting B below every drilled B	separation	highest lasting -0.0004, lowest drilled 0.061	held	signature
DP3	B rises with passes in U and F	monotone	monotone	held	signature
DP4	B ranks the loss	Spearman ≥ 0.7	0.93	held	signature
CT1	Competence lost, not slowed	U, F $< 15\%$ within 16,384 tokens	6.0, 11.7	held	break
CT2	A longer budget adds little for S	gain ≤ 10 over 4,096 tokens	54.2 to 75.8	failed	budget
Anchor (exploratory arms)					
AN1	The anchor protects	FA1 $\geq 40\%$ after AO, if the checks pass	48.2; parity check failed	not readable	anchor
AN2	Dose	FA1 > FA025 > F after AO; B in reverse	48.2, 27.9, 11.7	held	anchor
AN3	Replication on a second subset	U2 $\leq 20\%$, $B \geq 0.06$	6.5%, B 0.172	held	break
AN4	Dilution does not protect	FN $\leq 20\%$, if the checks pass	drilling check failed	not readable	anchor
Decision and competence					
DC1	The RL teacher skips the search later than trace checkpoints	later than R-P at 1.1, 3.3 and 7.7 passes	later than R-P at 3.3 and 7.7 passes, not at 1.1	failed*	decision
DC3	Forced, trace and RL checkpoints stay above their free accuracy and M0's forced	every such origin	not R-P-u140	failed*	decision
DC4	Forcing does not lift the solver above M0	R-S $\leq$ M0 forced after the flip	never evaluated: R-S never flips	not readable	decision
C-1	Forced accuracy holds through update 80	every origin	T-u30 falls to 21.4	failed	horizon
C-2	Trained checkpoints keep their margin over M0	every origin	T-u30 21.4, M0 39.3 at update 80	failed	horizon
C-3	Near zero by update 480, halving at least 10× later than the half-skip update	every origin	R-P-u140 halves at 10, half-skip at 1.7	failed*	horizon
E1	The coverage gap is a choice of mode	solver's never-solved count near the trace's	2 and 0 (M0 1)	held	earlier study
E2	Forced on held-out families, trained checkpoints near M0	close	58.9–62.2 against 54.2	held	earlier study
C2	The clone beats the teacher at update 80 in a rerun	by more than 10	41.9 against 53.9	failed	horizon
C3	The decision clocks repeat	half-skip in updates 3–6, within 1 of the first run	4.04, 4.81	held	decision
Repetition					
RR1	Drilled U and F stop searching sooner than S	both arms	U only	failed*	decision
RR-D	Competence lost at 7.7 passes, kept at 1.1	most lost; kept	-53.4, -58.6; at 1.1: -1.8, +3.9	held	break
RR-D2	The same at a third of the learning rate	below half of handoff	U 29.9 from 53.6, F 37.2 from 61.7	failed	stage
RR-D3	The same under IT	below half of handoff	U 34.9, F 36.7	failed	stage
RR-D3b	Under IT, free accuracy shows it	drilled lose ≥ 10, states at 1.1 passes ≤ 3	states at 1.1 passes lose 37.8–41.7	failed	decision
RR-D4	A cliff: most of the loss by update 2	F $< 30\%$ at update 2	45.1	failed	mechanism