Life of Research · Published 25 November 2026 · DOI 10.5281/zenodo.24537485

A Different Route Is Not a Different Destination: What Controlled On- Versus Off-Policy Distillation Rules Out About Forgetting, Sparsity and Generalisation, Why Its Null Does Not Carry Over to the Gap Between Fine-Tuning and Reinforcement Learning, and the Target-Matched Comparison That Would Assign the Credit

Pranay Mahendrakar
SONYTECH · ORCID 0009-0003-7224-029X
pranaymahendrakar@sonytech.in · mahendrakarpranay@gmail.com · +91 6361723454

Abstract. Reinforcement learning on a language model's own samples has been credited with three advantages over supervised fine-tuning: it forgets less of what the model already knew, it changes fewer parameters, and it generalises further. A September 2026 study tests the common cause behind all three, the use of on-policy data, in a setting built to isolate it. It distils a stronger teacher into a weaker student, crosses student-sampled against teacher-sampled trajectories with forward against reverse KL, and sweeps the learning rate. Rollout source barely moves final accuracy, forgetting or update sparsity; the learning rate moves forgetting and sparsity, and KL direction moves accuracy. In this analysis the author reviews and synthesises published results to show that this null is sound for what it varies and cannot be carried over to the fine-tuning versus reinforcement learning comparison it is read against. "On-policy" names at least seven separable properties, and the distillation design varies one of them: whose prefixes the loss is computed on. With a fixed teacher and a full-vocabulary loss, both arms share a single minimiser, the teacher's next-token distribution at every visited prefix, so the rollout source can change the route but not the destination. The retention evidence on the other side comes from comparisons in which the training target moves with the model along with the prefixes: supervised fine-tuning on samples the model itself produced, regenerated during training, forgets mildly or not at all, while the same kind of samples drawn once from the starting model forget badly. Because regenerating a fine-tuning sample changes prefixes and target together, that evidence implicates the pair rather than prefix source alone, and the distillation arms, whose teacher is frozen, never vary the target. The sparsity evidence is dominated by step size and numerical precision; off-policy preference methods are as sparse as on-policy ones in the checkpoints usually cited. The one on-policy benefit the distillation study does find, transfer to harder task variants, is the one imitation-learning theory predicts from prefix source alone. The paper consolidates the measured values in a table and a figure, sets out a property ledger and an attribution procedure, and specifies the target-matched and step-matched comparisons that would settle what remains open.

Keywords: on-policy learning; knowledge distillation; catastrophic forgetting; parameter-update sparsity; reinforcement learning from verifiable rewards; supervised fine-tuning; post-training; evaluation methodology

SONYTECH ORCID: 0009-0003-7224-029X

1. Introduction

Post-training a language model now usually combines three kinds of update: supervised fine-tuning on demonstrations, reinforcement learning against a verifier or reward model, and distillation from a stronger model (Ouyang et al., 2022; Hinton et al., 2015; Agarwal et al., 2023). A line of work since 2025 has argued that the second kind has properties the first lacks, and that one feature explains them: reinforcement learning trains on outputs sampled from the model being trained. Shenfeld et al. (2025) report that, at matched performance on a new task, reinforcement learning preserves earlier capabilities far better than fine-tuning, and attribute this to an implicit bias of on-policy updates toward solutions close to the starting model. Chen et al. (2025) reach a similar conclusion through a mixture-of-modes argument and ablations that rule out the KL penalty and the advantage estimator. Mukherjee et al. (2025) report that reinforcement learning updates only a small fraction of a model's parameters, and conjecture that training on data near the policy distribution is the reason. Chu et al. (2025) report that reinforcement learning generalises to unseen rule and visual variants where fine-tuning memorises. These results are now cited as reasons to prefer on-policy methods, including on-policy distillation, which keeps the teacher's dense supervision and replaces teacher-written trajectories with student-sampled ones (Agarwal et al., 2023; Gu et al., 2023; Song and Zheng, 2026).

Piskorz et al. (2026) challenge the common cause directly. Comparisons between fine-tuning and reinforcement learning change the objective, the reward signal, the density of supervision and the optimisation settings at once, so an effect attributed to on-policy data could come from any of them. Distillation offers a cleaner test: the trajectory source can be switched

© 2026 Pranay Mahendrakar. Licensed under CC BY 4.0.

between student and teacher while the loss, the teacher and the optimiser stay fixed. Across Llama and Qwen pairs and three reasoning tasks, they find no consistent advantage for student-sampled trajectories in final accuracy, forgetting or update sparsity. The learning rate dominates forgetting and sparsity; the direction of the token-level KL dominates accuracy and its stability. On-policy trajectories help in one place, transfer to harder variants of an arithmetic task, and that advantage does not reliably survive a later round of reinforcement learning.

The study is careful, and its central measurement is not in dispute here. What is in dispute is the reach of its conclusion. The authors write that their results make it difficult to attribute most of the observed differences between fine-tuning and reinforcement learning to rollout policy alone. That sentence does two jobs. Read narrowly, it says that in distillation the trajectory source is not what governs forgetting and sparsity, which the experiments support. Read broadly, it says the on-policy account of the fine-tuning versus reinforcement learning gap is weakened, which requires the distillation comparison to vary the same thing the on-policy account names. This paper argues that it does not.

In this analysis the author reviews and synthesises published results to show three things. First, "on-policy" covers at least seven separable properties of a training procedure, and the distillation design varies exactly one: the source of the prefixes at which the loss is evaluated. The properties the forgetting literature has actually manipulated, chiefly whether the training target is regenerated from the current model, are held fixed in both arms of the distillation study, at values the forgetting literature associates with forgetting. Second, a structural fact explains why rollout source should matter little for retention in that design. With a frozen teacher and a full-vocabulary per-token KL, both arms are driven toward the same distribution, the teacher's, at every prefix they visit; the arms differ in which prefixes are visited and in the path taken, not in where the loss would settle. Reinforcement learning without a KL penalty has a set of optimal policies, not a single one, and which member is reached depends on the path. Fine-tuning on a finite sample has yet another minimiser, the empirical distribution of that sample. These are not the same comparison. Third, the sparsity claim is mostly a measurement of step size at a numerical precision, as two independent groups found before the distillation study confirmed it from a third direction, and the generalisation result the distillation study does report is the benefit that imitation-learning theory predicts from prefix source specifically (Ross et al., 2010; Bengio et al., 2015).

The rest of the paper is organised as follows. Section 2 defines the seven properties. Section 3 summarises what the distillation study measured, with the values in Table 1. Section 4 sets out the minimiser argument. Sections 5, 6 and 7 take the three claimed advantages in turn: forgetting, sparsity and generalisation. Section 8 explains why a matched learning rate is not a matched step. Section 9 gives an attribution procedure as Algorithm 1, and Section 10 states what four accounts predict under five manipulations (Table 3). Sections 11 to 14 give the strongest case against this paper, what is not known, the studies that would settle it, and a conclusion. Figure 1 re-plots published sparsity and forgetting values against the variables this paper argues they track.

2. Seven Properties Under One Label

The word on-policy is used in the post-training literature for a procedure, not a property. A procedure is called on-policy when its training data come from the current model, and the label is then credited with whatever the procedure does. To ask what the label causes, the procedure has to be taken apart. The author distinguishes seven properties. The first six describe the update; the seventh describes how its effect is measured.

P1. Prefix source. The distribution over partial sequences at which the per-token loss is evaluated. In supervised fine-tuning and off-policy distillation, prefixes come from a dataset or a teacher; in reinforcement learning and on-policy distillation, from the student. This is the property Piskorz et al. (2026) vary, and the one the exposure-bias literature is about (Bengio et al., 2015; Ross et al., 2010).

P2. Target source. What the loss pushes each prediction toward. In supervised fine-tuning, the target is a stored token. In distillation, it is the teacher's next-token distribution. In policy-gradient reinforcement learning, there is no explicit per-token target; the update raises the probability of sampled sequences in proportion to their advantage, which reweights the current model's own distribution. With a KL penalty, the objective is equivalent to approximating a Bayesian posterior that updates the starting model by the evidence the reward provides (Korbak et al., 2022). In self-distillation, the target is the same model shown extra context (Shenfeld et al., 2026).

P3. Target tracking. Whether the target is recomputed from the current model during training or fixed in advance. Reinforcement learning's implicit target is recomputed at every step, because it is built from fresh samples. A teacher in distillation is usually frozen, so the target does not track the student no matter whose prefixes are visited. A fine-tuning set drawn once from the starting model is self-generated but does not track; one regenerated each epoch tracks approximately (Chen et al., 2025).

P4. Target form. A one-hot stored token, a full next-token distribution, a sampled-token estimate of a divergence, or a scalar reward on a whole sequence. Full-vocabulary distillation gives a soft target at every position, which a fine-tuning set does not (Hinton et al.,

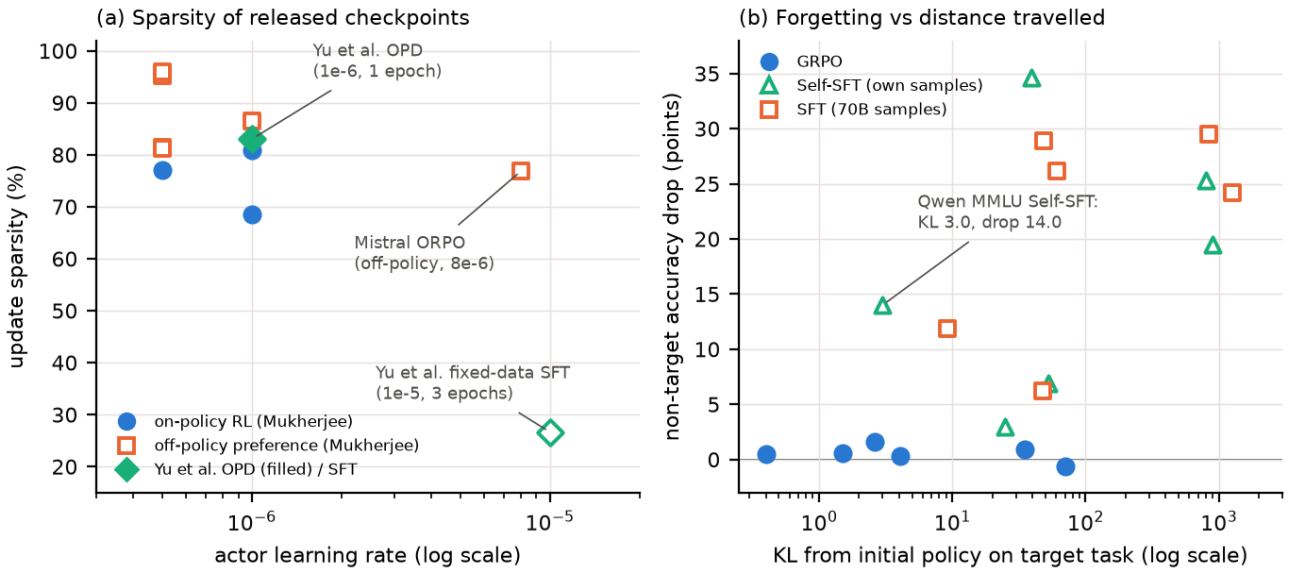


Figure 1. (a) Update sparsity (tolerance 1e-5, bfloat16) of released checkpoints against actor learning rate: sparsity from Mukherjee et al. (2025), Table 1; learning rates from Piskorz et al. (2026), Table 6; DeepSeek-R1-Zero omitted because its rate is undisclosed. Filled circles are on-policy, open squares off-policy preference methods; at 5e-7 the points 95.2 (DPO-70B) and 96.0 (KTO) overlap, as do 81.2 (KTO) and 81.4 (DPO-8B). Diamonds are the on-policy distillation and fixed-data fine-tuning runs of Yu et al. (2026), Tables 2, 6 and 7. (b) Mean accuracy drop on non-target tasks against KL from the initial policy on the target task, for Llama-3.2-1B and Qwen-2.5-1.5B on three target tasks, from Chen et al. (2025), Appendix Table 2.

2015; Kim and Rush, 2016).

P5. Negative signal. Whether the update can push probability away from outputs, as advantage-weighted policy gradients and contrastive preference losses can and maximum likelihood cannot (Tajwar et al., 2024).

P6. Divergence direction and step budget. The direction of the KL being minimised (forward is mode-covering, reverse is mode-seeking) and the distance the optimiser is allowed to travel: learning rate, number of steps, clipping, and any explicit KL penalty or trust region (Chan et al., 2021; Schulman et al., 2015; Schulman et al., 2017). These two are grouped because they jointly fix how far and in what direction a run moves from its start for a given amount of training.

P7. Measurement. How retention, sparsity and generalisation are read: which unrelated benchmarks define forgetting, which threshold and floating-point format define an unchanged parameter, and whether comparisons are made at a matched learning rate, a matched new-task score, or along a Pareto frontier over hyperparameters.

Table 2 places the procedures that appear in this debate on the first four properties, together with the minimiser each would approach given unlimited capacity and training, a notion Section 4 develops. The table is the author's classification; the retention column reports what the cited studies found.

Read down the "target tracks model?" column, the rows with good retention are those in which the target moves with the model or sits close to the start by construction. Read down the "prefixes" column, the pattern is weaker: Self-SFT trains on the model's own outputs and retains poorly, and the oracle row trains on labels the model did not produce and retains best. The two distillation rows differ only in the prefix column. That is the contrast the distillation study runs, and it is not the contrast the retention rows above it support.

3. What the Controlled Distillation Study Reports

Piskorz et al. (2026) distil task-specific Llama-3.1-8B teachers into Llama-3.2-1B-Instruct students, with Qwen2.5-7B into Qwen2.5-1.5B as a second family, on medical question answering, scientific reasoning and the Countdown arithmetic game. Each teacher is fine-tuned on its task with low-rank adapters and, for two of the three tasks, refined with GRPO, so that it is clearly stronger than the student. Students are fully fine-tuned for 150 optimiser steps with 32 prompts per step and three seeds per configuration. The loss is the token-level KL between student and teacher computed over the whole vocabulary, in either direction, averaged over positions of a trajectory drawn from either the teacher (off-policy) or the student (on-policy). The training details state that one completion is sampled for every prompt selected into a batch. Teacher distributions are sharpened with temperature 0.5. Forgetting is the change in the mean score over seven unrelated benchmarks, and sparsity is the fraction of parameters whose absolute change stays below 1e-6, following Mukherjee et al. (2025).

The headline results, collected with the other measured values in Table 1, are these. The best mean held-out

Table 2. Procedures in the on-policy debate placed on prefix source, target source and target tracking, with the minimiser each approaches in the capacity limit and the retention each cited study reports. Classification by the author.

Procedure	Prefixes	Per-token target	Target tracks model?	Minimiser (capacity limit)	Retention reported
SFT on external samples	dataset	stored token	no	empirical sample	poor (Chen et al., 2025)
Self-SFT, sampled once	own, initial	stored token	no	empirical sample	poor (Chen et al., 2025)
Iterative-SFT, per epoch	own, recent	stored token	approx.	moving	mild to none (Chen et al., 2025)
1-0 Reinforce	own, current	reweighted self	yes	set of correct policies	good (Shenfeld et al., 2025)
GRPO, no KL penalty	own, current	reweighted self	yes	set of correct policies	good (Shenfeld et al., 2025)
SFT on KL-minimal oracle	oracle	stored token	no	oracle (near start)	better than RL (Shenfeld et al., 2025)
Off-policy distillation	teacher	teacher dist.	no (frozen)	teacher	LR-governed (Piskorz et al., 2026)
On-policy distillation	own, current	teacher dist.	no (frozen)	teacher	LR-governed (Piskorz et al., 2026)
Self-distillation (SDFT)	own, current	self + context	yes (EMA)	moving, near self	good (Shenfeld et al., 2026)

Table 1. Measured values the argument rests on, one row per published result, with the source on each row. "pts" are percentage points; sparsity is the percentage of parameters whose change falls below the stated threshold.

Source	Setting	Comparison	Reported value
Piskorz et al. 2026	Llama 8B to 1B, 3 tasks	best mean ID accuracy, on vs off	72% vs 73%
Piskorz et al. 2026	same	unrelated-benchmark change, LR 1e-5 / 5e-5	at most 1.3 pts / -11.2 to -14.0 pts
Piskorz et al. 2026	same, threshold 1e-6	sparsity, LR 1e-5 / 5e-5	85.3-89.6% / 51.9-60.0%
Piskorz et al. 2026	Countdown-4E	pass@k, student-favoured vs teacher-favoured	10-15% higher
Piskorz et al. 2026	public checkpoints	median actor LR, SFT vs RL/preference	5e-6 vs 5e-7
Chen et al. 2025	Llama 1B, IFEval target	non-target drop, Self-SFT / SFT / GRPO	6.9 / 26.2 / 1.6 pts
Chen et al. 2025	same	KL from start, Self-SFT / SFT / GRPO	52.4 / 61.1 / 2.6
Chen et al. 2025	Qwen 1.5B, MMLU target	Self-SFT drop at KL 3.0	14.0 pts
Shenfeld et al. 2025	ParityMNIST / LLM	forgetting vs KL fit	R^2 0.96 / 0.71 (quadratic)
Mukherjee et al. 2025	released checkpoints, tol. 1e-5	sparsity, off-policy pref. vs on-policy RL	76.9-96.0% vs 68.5-86.0%
Mukherjee et al. 2025	Qwen2.5-Math-7B	sparsity, SFT on own rejection samples	91.2%
Mukherjee et al. 2025	Llama-3.1-8B-SFT, Zephyr	sparsity, DPO on unadapted data	6.8%, 7.7%
Yu et al. 2026	R1-Qwen 1.5B	sparsity, OPD (1e-6, 1 ep.) vs SFT (1e-5, 3 ep.)	83.04% vs 26.63%
Yu et al. 2026	JustRL checkpoint	sparsity of an RLVR update	34.68%
Chen et al. 2026	Qwen2.5-7B, TRACE	backward transfer, SFT vs on-policy replay 10% / 1%	-13.93 vs -0.65 / -2.29

accuracy across the three tasks is 72 percent for on-policy and 73 percent for off-policy distillation. Forward KL reaches 71 to 73 percent in every rollout and learning-rate combination, while reverse KL ranges from 35 to 72 percent and is sensitive to the learning rate. At a learning rate of 1e-5 the mean score on unrelated benchmarks moves by at most 1.3 points; at 5e-5 it falls by 11.2 to 14.0 points. Sparsity is 85.3 to 89.6 percent at the lower rate and 51.9 to 60.0 percent at the higher, and the off-policy arm is at least as sparse as the on-policy arm in every matched comparison. A sweep of six learning rates on Countdown shows forgetting rising and sparsity falling steadily with the rate in both arms, and at matched rates the off-policy arm forgets slightly less in nearly every comparison.

A second experiment interpolates and extrapolates the trajectory source along a student-teacher spectrum. Forward KL keeps held-out Countdown accuracy above 80 percent across the spectrum, varying by 5.2 points at the lower learning rate; reverse KL benefits from student-favoured trajectories and is unstable when they are teacher-favoured. A gradient analysis explains the asymmetry: the forward-KL gradient with respect to the student's logits is the difference between the two distributions, which is bounded and changes smoothly with the trajectory distribution, while the reverse-KL gradient is weighted by the student's own probabilities and by a log-ratio that can be very large. Forgetting and sparsity again move little along the spectrum.

Three further results matter for what follows. Checkpoints trained on more student-like trajectories score 10 to 15 percent higher pass@k on a harder Countdown variant, under both KL directions. When those checkpoints are trained further with reinforcement learning on the harder task, off-policy checkpoints end highest despite starting lower, a pattern the authors relate to the finding of Zhang et al. (2026a) that the best fine-tuned checkpoint is not always the best start for reinforcement learning. And when the teacher is told to reason in Spanish, every configuration except on-policy

reverse KL transfers that style almost completely to the student. The results survive removing gradient clipping, replacing the full-vocabulary loss with sampled estimators, and a single-run experiment on longer mathematical reasoning, in which on-policy reverse KL reaches the highest score at the lower learning rate, a result the authors treat as suggestive.

4. One Destination, Two Routes

The argument of this section is a derivation, not a measurement, and is stated as such. Consider the per-token loss Piskorz et al. (2026) minimise: at each prefix visited by the rollout distribution, the KL between the student's and the teacher's next-token distributions over the full vocabulary. Whichever direction is used, the KL at a prefix is zero exactly when the two distributions coincide there and positive otherwise. The total loss is an average of non-negative terms weighted by how often each prefix is visited. A student that matched the teacher at every visited prefix would therefore attain zero loss in both arms and both directions, and no other student would. In the limit of sufficient capacity and training, both arms approach the same object: the teacher's conditional distribution on the set of prefixes visited.

Two qualifications follow, and they are where the rollout source can still matter. The first is support. Each arm constrains the student only where it visits; the off-policy arm visits prefixes the teacher writes, the on-policy arm prefixes the student writes, including those that have already gone wrong. On prefixes neither arm visits, nothing in the loss pins the student down. The second is the path. With finite capacity, steps and learning rate, the student settles at a compromise, and the weights on prefixes, the step sizes and the divergence direction determine which compromise. Wu et al. (2024) make the related point for the two directions of KL: in their analysis and experiments, forward and reverse KL share an optimisation objective and converge after enough epochs, and differ in what they emphasise early, the head of the teacher's distribution for forward KL and the tail for reverse KL. Di et al. (2026) show where the shared-destination property breaks: when a student is distilled sequentially from several teachers, forward KL targets a weighted arithmetic mixture of them and reverse KL a normalised geometric aggregate, so the two directions then have different destinations. With one frozen teacher, as in the distillation study, they do not.

Set this beside the other two objectives in the debate. A model fine-tuned by maximum likelihood on a fixed set of responses approaches, in the same limit, the empirical distribution of that set: all probability on the stored responses for each prompt, in proportion to how often each appears. If the set was sampled from the starting model, every response in it was plausible to that model, but the empirical distribution of at most five retained samples per prompt, the number Chen et al. (2025) draw, is a much narrower object than the distribution that produced them. A forgetting predictor that compares the trained model with the starting one, such as the KL used by Shenfeld et al. (2025), will register that narrowing even though each training sequence was the model's own. This is consistent with Chen et al. (2025), whose Self-SFT arm trains on the starting model's correct samples and, for Llama-3.2-1B trained on IFEval, ends at a KL from the start of 52.4, against 2.6 for GRPO (Table 1).

Reinforcement learning with a binary reward and no KL penalty, the setting Shenfeld et al. (2025) use for their main comparisons, has no unique minimiser at all: every policy that puts all its probability on correct outputs is optimal. Which of these a run reaches is decided by the path, and the RL's Razor claim is precisely a claim about the path, that sampling from the current model and reweighting what it already produces selects the optimal policy nearest the start in KL. With a KL penalty the objective acquires a single target, which Korbak et al. (2022) identify with a Bayesian posterior having the starting model as its prior, and which therefore stays anchored to the start by construction. In neither case is there an external target that the procedure is driven toward regardless of where it samples.

The consequence for the distillation null is direct. In a design whose two arms share a destination, the trajectory source can affect retention only through support and path. Support matters for behaviour on prefixes the student reaches, which is the generalisation question taken up in Section 7, and has no obvious bearing on unrelated benchmarks, whose inputs neither arm visits. Path matters through how far the student travels toward the teacher before training stops, which is set by the learning rate and the number of steps far more than by the trajectory source. That is what Piskorz et al. (2026) observe. The observation is a sound finding about distillation toward a fixed teacher, and it is what the minimiser structure predicts. It does not bear on the claim that reinforcement learning retains because its implicit target tracks the model, because neither distillation arm has a target that tracks the student.

The SDFT procedure of Shenfeld et al. (2026) is the instructive exception, because it is distillation whose target does track. The teacher is the student itself, conditioned on a demonstration, with weights kept as an exponential moving average of the student's. Its authors note in the caption of their first figure that conditioning on demonstrations gives a teacher whose output distribution is substantially closer to the base model while keeping new-task accuracy. In their comparison on the Tool Use task, which swept learning rate, batch size and epochs for every method and selected on validation performance, offline distillation from the same teacher on a fixed set of teacher outputs improved on fine-tuning but stayed below the on-policy procedure in new-task performance; that comparison does not measure retention. That comparison changes three properties at

once relative to the Piskorz design: the target sits near the student, it moves with the student, and the offline arm reuses a fixed set. It is therefore evidence for the bundle, not for prefix source alone, but it is the kind of evidence the distillation null does not reach.

5. Forgetting: Three Accounts and the Arm Each Needs

The retention advantage of reinforcement learning has three published explanations, and each needs a different comparison to test it.

The distance account. Shenfeld et al. (2025) find that forgetting is predicted by the forward KL from the starting policy to the trained one, evaluated on the new task's inputs, across methods and hyperparameters, with R^2 of 0.96 on a ParityMNIST task and a quadratic fit with R^2 of 0.71 in their language-model experiments. They then construct, for the toy task, an oracle labelling that reaches perfect accuracy at minimal KL from the starting model, and report that fine-tuning on it forgets less than reinforcement learning. They also report that, in the same toy setting, fine-tuning on outputs generated by a reinforcement-learned model matches that model's accuracy-forgetting trade-off. On this account, on-policy sampling matters because it is one way of reaching a nearby solution, not because sampling as such protects anything. Their comparison of four objectives supports the sampling route against the negative-signal route: the two on-policy procedures, GRPO and 1-0 Reinforce, which samples from the current model and fine-tunes on the correct samples only, retain better than the two offline ones, fine-tuning and SimPO, whether or not negative examples are used.

The data account. Chen et al. (2025) find, across Llama and Qwen models and three target tasks, that GRPO forgets less than fine-tuning on either a 70-billion-parameter model's outputs or the starting model's own filtered outputs, and trace the difference to the mode-seeking character of reverse-KL updates on a multi-modal starting distribution. Their ablations remove the KL penalty and replace the advantage estimator without restoring forgetting. The decisive comparison for the present argument is among their three fine-tuning variants. Training on the starting model's own correct samples, generated once, forgets severely. Training on samples regenerated at the start of each epoch, which they call Iterative-SFT, reaches comparable or higher target accuracy with mild to no forgetting, and fine-tuning on traces produced during a reinforcement-learning run also reduces forgetting. What separates Self-SFT from Iterative-SFT is how current the self-generated samples are, which the authors describe as a degree of on-policy-ness. In fine-tuning the stored response is both the sequence whose prefixes are scored and the target, so regenerating it from the current model changes prefix source (P1) and target tracking (P3) together, with only target form (P4) held fixed. The third variant also argues against a pure tracking reading: traces collected during a reinforcement-learning run are fixed before the fine-tuning starts, so their target does not track in the sense of P3, yet SFT on them "trails full RL marginally in terms of gains but exhibits only slightly larger forgetting" (Chen et al., 2025, Appendix A.4.1). The implementation details give one learning rate per model size, 1e-4 for the two small models, with no separate value listed for GRPO, so this comparison is not explained by a learning-rate difference between the fine-tuning arms.

The step account. Piskorz et al. (2026) find, for distillation, that the learning rate governs forgetting and that forgetting is not a necessary cost of reaching high task accuracy: with forward KL and a small learning rate, students reach top accuracy with no measurable loss on unrelated benchmarks. Rofin et al. (2026) show for fine-tuning that models trained to the same loss with large and small steps end in qualitatively different places, and connect this to the heightened sensitivity of long-pretrained models to modification described by Springer et al. (2025). Kalajdzievski (2024) finds that forgetting under low-rank fine-tuning grows as a shifted power law in the number of update steps and parameters tuned. Biderman et al. (2024) find that low-rank adaptation both learns less and forgets less than full fine-tuning. Chen et al. (2025) themselves report a learning-rate trade-off for Self-SFT: a high rate is needed to reach high target accuracy and brings severe forgetting, while a small rate reduces forgetting but does not reach the same accuracy even with more epochs.

These accounts are compatible once each is assigned the property it manipulates. The distance account is about where training ends up relative to the start; the data account shows that one reliable way to end up nearby is a target that is rebuilt from the current model; the step account shows that, for a fixed target, how far one goes along the path is controlled by the step budget. The distillation design fixes the target far from the student, since the teacher is a different, larger model trained further on the task, and gives it no way to move. Within that design only the step account can act, and it does.

A wider set of results points the same way. Fine-tuning on a model's own rewrites of the training data reduces forgetting relative to fine-tuning on the original labels (Yang et al., 2024). Fine-tuning on model-generated responses retains other reasoning skills better than fine-tuning on human ones, a difference Ren et al. (2024) relate to the model's familiarity with its own outputs, measured as lower perplexity before training. Wu et al. (2025a) trace a similar retention benefit to the absence of high-perplexity tokens in model-generated sequences and recover most of it by masking such tokens in ground-truth data. Mixing self-generated dialogues into task data, without changing the optimiser or schedule, keeps performance close to the original model across 50 evaluation scenarios and gives the best result in 40 of them, which its authors connect to parameter drift

induced by style (Sun et al., 2026). Replaying the latest checkpoint's own filtered outputs on old prompts improves backward transfer on a continual benchmark from -13.93 with no replay to -0.65 at a 10 percent replay budget, against -1.37 for replaying gold data from earlier tasks at the same budget. Its authors find that replaying the model's own low-scoring outputs instead (-3.47) does worse than gold replay, and conclude that "On-policy sampling alone is insufficient—the high-reward filter is the decisive factor" (Chen et al., 2026). For small models, acquisition and forgetting both track the post-trained model's distribution shift linearly, and an in-distribution rewrite produced by a slightly larger model of the same family does better than off- and on-policy distillation (Koh et al., 2026). Trajectory-mixed supervision from the model's own earlier checkpoints shifts the accuracy-retention frontier, with drift between policy and labels predicting forgetting, although reinforcement learning remains best for retention there (Khan et al., 2026). A theoretical study of classification tasks shows that, starting from a policy with no stylistic preference within a semantic class, exact policy gradients keep that symmetry while fine-tuning on a stylistically biased teacher drifts along it, producing forgetting that reinforcement fine-tuning avoids (Liang et al., 2026). Within on-policy distillation itself, asymmetric on-policy distillation, which replaces negative reinforcement with local divergence minimisation where the teacher-derived advantage is not positive, reports better capability retention during sequential tool-use adaptation than the standard recipe (Jia et al., 2026); there the trajectory source is the same in both arms and the update form differs. Every one of these manipulates the target's source, its form or its distance from the model. None manipulates prefix source with the target held fixed.

The distillation study's style experiment is a direct instance. A teacher instructed to reason in Spanish is a target that differs from the student along one dimension, and in three of the four configurations the student adopts it almost entirely. That is what a fixed, distant target does under a loss that drives the student toward it. The fourth configuration, on-policy reverse KL, preserves English, and the authors suggest the teacher still supports English continuations on English student prefixes. That is a support effect in the sense of Section 4: on prefixes the student writes, the teacher's distribution contains an English mode the mode-seeking loss can match.

None of this means reinforcement learning cannot forget. InstructGPT's PPO training on its API distribution produced regressions on several public benchmarks, which its authors call an alignment tax and mitigate by mixing in pretraining updates (Ouyang et al., 2022). Lin et al. (2023) document a pronounced alignment tax for RLHF on a 3-billion-parameter model and find that averaging pre- and post-RLHF weights gives the best trade-off among the methods they compare. Luo et al. (2026) report that standard reinforcement learning forgets on a continual benchmark built from recent multimodal reasoning tasks, and trace this to a KL penalty evaluated on current-task data when the damage is on earlier tasks. Chen et al. (2025) show in simulation that reverse-KL updates also forget once the target mode is far enough from the starting modes. At the mechanistic level, Rojas Nunez et al. (2026) find that reinforcement learning preserves more of a base model's circuits than fine-tuning on a scientific question-answering task, and that it adapts more slowly; slower adaptation is what a smaller effective step would also produce. Lai et al. (2025) report that reinforcement fine-tuning preserves prior tasks in continual post-training of a vision-language model, and find the stability not to come primarily from the KL penalty or from chain-of-thought.

The opposite caution also applies. Off-policy updates can retain well when their step is controlled. Wu et al. (2026) learn a fine-tuning update on an earlier checkpoint and apply a scaled version of it to the post-trained model, and report that this Pareto-dominates both fine-tuning and on-policy self-distillation in new- and old-task performance, without on-policy sampling. Proximal SFT bounds the drift of each fine-tuning update with a trust-region objective and reports better out-of-domain behaviour than plain fine-tuning (Zhu et al., 2025b). These results fit the step account and the distance account together: what protects old capabilities is not being moved far, whatever generated the data.

6. Sparsity Is a Measurement of Step Size at a Precision

The sparsity claim has a measurement problem that the forgetting claim does not, and it has been noticed three times.

Mukherjee et al. (2025) measure update sparsity on released checkpoints in bfloat16, counting a parameter as unchanged when its before and after values differ by no more than 1e-5. They report 68.5 to 96.0 percent sparsity after reinforcement learning across ten checkpoints and seven algorithms, against 6 to 15 percent after fine-tuning, and conjecture that the cause is training on data near the current policy, with KL penalties and gradient clipping having little effect. Their own Table 1 marks four of the algorithms as off-policy: DPO, KTO, SimPO and ORPO, all trained on fixed preference data. These checkpoints are not less sparse than the on-policy ones; they span 76.9 to 96.0 percent against 68.5 to 86.0 percent for GRPO, PPO, PRIME and R1-Zero (Table 1, Figure 1a). Their control experiment points to the same reading. Fine-tuning on the model's own rejection-sampled outputs gives 91.2 percent sparsity, iterative rejection-sampling fine-tuning 69.4 percent, and DPO on data the model had not first been adapted to only 6.8 and 7.7 percent. "In-distribution data", as Mukherjee et al. define it, is a property of the

target's distance from the model (P2 and P3), not of prefix source, and their own evidence shows it producing sparsity under maximum likelihood.

Shenfeld et al. (2025), testing the sparsity hypothesis as a possible predictor of forgetting, report that in their setting the sparsity came from training in bfloat16: small updates fail to change a value whose limited mantissa cannot represent them, and the same training in float32 gave models with identical performance and no sparsity. Zhu et al. (2025a) give a three-part account in which a KL anchor limits the update, the model's geometry steers it away from principal directions into low-curvature subspaces, and limited precision hides the resulting small changes, so that an off-principal bias appears as sparsity. They also report that fine-tuning, by contrast, targets principal weights and distorts the spectrum. On both accounts, sparsity is what small, well-directed updates look like at a coarse precision.

Piskorz et al. (2026) add the step-size side. In their controlled runs, sparsity falls roughly linearly as the learning rate rises, from 85 to 90 percent at 1e-5 to 52 to 60 percent at 5e-5, with little dependence on the trajectory source. Their audit of the checkpoints Mukherjee et al. analysed finds a median actor learning rate of 5e-7 for the reinforcement-learning and preference checkpoints against 5e-6 for the fine-tuning ones (their Table 6). The learning rate is not a complete explanation: in Figure 1a, the ORPO checkpoint, trained at 8e-6 from a base model with an off-policy, reference-free objective, still shows 76.9 percent sparsity. But neither is the on-policy label, which does not separate the points at all.

The most recent on-policy sparsity result shows how easily the comparison is confounded. Yu et al. (2026) report that on-policy distillation updates are small and sparse, distributed across layers, and biased away from the source model's principal coordinates, and they offer this as evidence that on-policy distillation carries weight-space signatures of on-policy post-training despite dense teacher supervision. In their matched comparison, the on-policy run has 83.04 percent sparsity against 26.63 percent for fine-tuning on fixed teacher outputs. Their configuration tables show the on-policy run at a learning rate of 1e-6 for one epoch and the fine-tuning reference at 1e-5 for three epochs. The comparison therefore varies prefix source and target form together with a tenfold difference in learning rate and a threefold difference in passes. Their own table also contains a reinforcement-learning checkpoint, JustRL, whose update is 34.68 percent sparse, which they describe as atypically large and dense. A sparse student update under on-policy distillation from that dense teacher is consistent with the step account and says nothing about on-policy data as such.

Figure 1 re-plots the published values on the axes this paper argues they track. Panel (a) places the nine checkpoints of Mukherjee et al. (2025) whose learning rate is disclosed, and the two runs of Yu et al. (2026), on actor learning rate against update sparsity. Panel (b) places the eighteen model-task-method cells of Chen et al. (2025) on KL from the initial policy against forgetting.

Panel (a) shows that the on-policy and off-policy checkpoints are interleaved, and that the one large gap, between the two Yu et al. runs, coincides with a tenfold change in learning rate. Panel (b) shows GRPO clustered at low forgetting across two orders of magnitude of KL, and the two fine-tuning variants spread at high forgetting. It also shows the limits of the distance account in that dataset: the Qwen Self-SFT cell trained on MMLU loses 14.0 points at a KL of 3.0, while two GRPO cells sit at KL 34.9 and 70.6 with drops under one point. Chen et al. (2025) title the corresponding appendix as showing that KL and forgetting correlate moderately, and the panel bears that out. KL measured on the target task's inputs is a summary of how far the model moved, not a complete account of where.

7. Generalisation: Where Prefix Source Should Matter

If the trajectory source matters anywhere, it should matter for behaviour on prefixes the student reaches and the teacher would not. That is the classical case for on-policy imitation. A policy trained only on expert states meets, at test time, states produced by its own errors, and its mistakes compound; training on states the learner itself visits, labelled by the expert, removes the mismatch and yields guarantees under the learner's own state distribution (Ross et al., 2010). Scheduled sampling made the same move for sequence models (Bengio et al., 2015), and on-policy distillation for language models was introduced on this rationale (Agarwal et al., 2023; Gu et al., 2023). The prediction is specific: the on-policy advantage should appear where the student's own trajectories leave the teacher's support, as in longer or harder problems, and should be small where the student stays close to teacher-like trajectories anyway.

The distillation study's one clear on-policy benefit fits this prediction. Students trained on more student-like trajectories transfer better to a harder Countdown variant, by 10 to 15 percent in pass@k under both divergence directions, and the single-run longer-reasoning experiment points the same way (Piskorz et al., 2026). This is a prefix-source effect of the kind the support argument in Section 4 allows, and it is consistent with Speculative Knowledge Distillation's motivation that fixed teacher data mismatches the student's inference-time distribution (Xu et al., 2024b). The same literature records the cost side: on student prefixes the teacher may be out of its own distribution and its feedback unreliable (Xu et al., 2024b), and conditioning on student-generated prefixes is one of three failure mechanisms Zhu et al. (2026) identify for on-policy

distillation in mathematical reasoning. Li et al. (2026) find that successful on-policy distillation aligns the student with the teacher on a small set of high-probability tokens at student-visited states, holding 97 to 99 percent of the probability mass, and that same-family teachers of 1.5 and 7 billion parameters are indistinguishable from the student's perspective, so a teacher must offer something the student has not already seen. Armandpour et al. (2026) find distillation gradients better aligned with an ideal success-maximising update on incorrect rollouts than on correct ones. All of these concern what happens on student-visited states, which is the territory prefix source governs.

The generalisation evidence for reinforcement learning over fine-tuning, by contrast, mixes prefix source with target form and data design. Chu et al. (2025) find that outcome-rewarded reinforcement learning generalises to unseen rule and visual variants of a card game and a navigation task where fine-tuning memorises, and that fine-tuning is still needed to stabilise output format. Lin et al. (2025) report that much of that fine-tuning failure comes from training on fixed instruction templates: adding prompt diversity restores generalisation to instruction variants, chain-of-thought supervision helps with harder variants, and the two together match or exceed the reinforcement-learning baselines on their benchmarks. Huan et al. (2025) find that, on math-only data with Qwen3-14B, reinforcement learning transfers across domains while fine-tuning often erodes general capabilities, with larger representation and output drift for fine-tuning. Yuan et al. (2025) find that reinforcement learning teaches compositions of previously learned string functions that transfer to new tasks, while next-token training on the same data does not. Matsutani et al. (2025) find that reinforcement learning compresses incorrect reasoning trajectories while fine-tuning expands correct ones. Yue et al. (2025) find that reinforcement learning sharpens what the base model already samples while distillation can introduce reasoning patterns from the teacher. These comparisons vary the target (stored teacher traces versus reweighted self-samples), the negative signal and, in several cases, data diversity, along with prefix source.

A group of methods tries to obtain reinforcement learning's generalisation without sampling sequences. One-token rollout samples candidate next tokens from the current policy at each position and rewards those matching the stored token, and its authors read its gains as evidence that the on-policy nature of data drives generalisation (Ming et al., 2025). In-distribution fine-tuning modifies the loss using a measure of how well data match the model-induced distribution, and a companion decoding method rewrites the training corpus toward that distribution (Zhang et al., 2026b). Dynamic fine-tuning rescales each token's loss by its probability, which its authors derive as a correction to an implicit reward in the fine-tuning gradient (Wu et al., 2025b). Proximal SFT bounds drift with a trust region (Zhu et al., 2025b). None of these samples whole trajectories from the current policy during training. What they change is the target, or how strongly the update pushes toward targets the model finds unlikely, which is a target-distance and step-size intervention. Their reported gains therefore bear on P2, P4 and P6, not on P1, whatever their authors call them.

The reversal after reinforcement learning deserves a separate note. The on-policy checkpoints that generalised better to the harder Countdown variant did not end best after reinforcement learning on that variant; off-policy checkpoints distilled with forward KL at the lower learning rate, or with reverse KL at the higher one, started near zero accuracy, climbed steadily and finished highest, while reverse-KL checkpoints distilled at the lower learning rate rose quickly and then collapsed (Piskorz et al., 2026). Zhang et al. (2026a) report the same kind of reversal for fine-tuning: stronger fine-tuned checkpoints can underperform weaker ones after identical reinforcement learning, which they attribute to a mismatch between the distribution that generated the fine-tuning data and the policy being optimised. One reading consistent with both is that a starting point's value for later reinforcement learning depends on the breadth of what it can still sample, which forward KL and off-policy coverage preserve, more than on its immediate accuracy. That reading is a hypothesis; neither study measured coverage at the start of the reinforcement-learning stage in a way that tests it.

Task accuracy, and the preference-learning precedent. The distillation study finds no consistent on-policy advantage in final in-distribution accuracy. The preference-learning literature, where online and offline methods were compared earlier and at larger scale, found the opposite, and it is worth asking why. Tajwar et al. (2024) find that methods using on-policy sampling or a negative gradient generally beat offline maximum-likelihood objectives, and unify the two under mode-seeking behaviour that moves probability between categories faster than maximum likelihood can. Tang et al. (2024) find a clear advantage for online over offline alignment in the context of reward over-optimisation, rule out offline data coverage and data quality on their own as explanations, and observe that offline training produces policies better at pairwise classification and worse at generation, while online training does the reverse. Xu et al. (2024a) find that a well-tuned PPO beats DPO across their dialogue and code benchmarks. In all three, the online arm's target is a reward evaluated on the policy's own samples, and the offline arm's target is a fixed set of preference pairs, so prefix source, target tracking and the negative signal change together. The distillation study's null on accuracy is consistent with them once that is noted: when the target is a full teacher distribution available at every prefix, as in the full-vocabulary distillation loss, the offline arm already

receives at every visited position the dense, graded information that an online preference method has to obtain by sampling. Prefix source then has less to add to in-distribution accuracy. This is an interpretation by the author; none of these studies ran the crossed design.

8. A Matched Learning Rate Is Not a Matched Step

The distillation study's most transferable lesson is methodological: compare at matched optimisation settings, because the learning rate moves forgetting and sparsity more than the factor under study. Its comparisons hold the learning rate fixed within a single loss family, where the same learning rate does mean roughly the same step. Carrying the lesson to fine-tuning against reinforcement learning requires care, because a matched nominal learning rate is not a matched step across losses.

A policy-gradient update multiplies each sequence's log-likelihood gradient by a normalised advantage. With a binary reward and group normalisation, groups in which all samples succeed or all fail contribute nothing, and the remaining sequences receive weights of both signs that partly cancel. A maximum-likelihood update on a stored set pushes every sequence up with weight one, and its gradient is largest exactly on the tokens the model finds least likely, which are the tokens Wu et al. (2025a) find responsible for much of fine-tuning's collateral damage. A full-vocabulary forward-KL update has a per-logit gradient bounded by one in magnitude (Piskorz et al., 2026). Gradient clipping, which the distillation study applies throughout and finds active at every logged step of its Countdown runs, further decouples the nominal learning rate from the distance travelled. Chen et al. (2025) used one learning rate per model size for all methods and still found a large retention gap between GRPO and both fine-tuning variants, which already shows that equalising the nominal rate does not equalise the effect.

The estimator adds a further gap between what an objective is said to be and what an update does. Tang and Munos (2025) show that several common implementations of KL gradients in language-model reinforcement learning differentiate through a KL estimate in a way that does not yield the intended gradient, or account for only part of the sequential dependence. On-policy distillation is usually implemented with exactly such estimators: a sampled-token log-ratio standing in for reverse KL, with gradients stopped through the sampling. The distillation study sidesteps much of this by computing the KL over the whole vocabulary and checks that sampled estimators do not change its conclusions, but comparisons elsewhere that pair an on-policy sampled estimator with an off-policy full-distribution loss change the update's form along with the trajectory source. When two arms differ in estimator, they differ in P4 and P6 as well as P1.

The quantity a step-matched comparison should hold fixed is the distance travelled in the space where forgetting is measured, not a hyperparameter. Three candidates are available in the published work: the KL from the starting policy on target inputs (Shenfeld et al., 2025; Chen et al., 2025), the norm of the weight update relative to the weights (Yu et al., 2026), and the position on a Pareto frontier of new-task score against old-task score across many hyperparameter settings (Shenfeld et al., 2025). The last is the most informative, because it compares the best each method can do rather than one setting of each. RL's Razor reports its central comparison as Pareto frontiers over dozens of hyperparameter settings for both methods, which is the strongest available answer to the learning-rate objection for the fine-tuning versus reinforcement learning gap. The distillation study reports two learning rates for most comparisons and a six-point sweep on one task, and does not construct frontiers; its finding that forgetting is not a necessary cost of accuracy under forward KL at a small learning rate is, in frontier terms, a statement that both distillation arms reach the favourable corner.

9. An Attribution Procedure

The author proposes the following procedure for reading a claimed on-policy benefit. It is stated as a decision structure; its first steps classify the claim and the comparison, and its later steps decide which property the evidence can be credited to.

Applied to the main results discussed here, the procedure assigns them as follows. The distillation study's null on forgetting and sparsity is credited to P1 under a shared minimiser, and is informative about P1 alone (step 4). Its harder-variant gain is credited to P1 under step 7. The retention gap of Chen et al. (2025) between Self-SFT and Iterative-SFT is credited to the bundle of P1 and P3 (step 3), because regenerating a stored response changes both, and their GRPO-versus-fine-tuning gap to the bundle of P1, P2, P3, P5 and P6 (step 3). The RL's Razor oracle result is credited to P2, the distance of the target from the start. Yu et al.'s sparsity contrast is step-confounded (step 5). Mukherjee et al.'s sparsity table, read with the learning rates of Piskorz et al., is step-confounded in part and does not separate on P1 in any case.

10. What Four Accounts Predict

Table 3 states, for four accounts of the on-policy effects, the outcome each predicts under five manipulations. The entries are predictions derived by the author from each account as published; none is a measured value. A1 is the prefix-source account (sampling one's own trajectories protects and generalises). A2 is the target-tracking account (a target rebuilt from the current model keeps training near the start). A3 is the step account (distance travelled governs forgetting and sparsity). A4 is the precision account, which applies to sparsity only.

Algorithm 1. Attributing a reported on-policy benefit

```
Input: a claim C ("on-policy training improves outcome Y")
       and the comparison E offered for it.
1. Classify Y: retention on unrelated inputs | in-distribution accuracy
   | transfer to harder or longer instances | update sparsity | style.
2. For each arm of E, record P1 prefix source, P2 target source,
   P3 target tracking, P4 target form, P5 negative signal,
   P6 divergence direction and step budget, P7 measurement.
3. List the properties that differ between the arms.
   If more than one differs, credit C to the bundle, not to P1.
4. Name the capacity-limit minimiser of each arm.
   If the minimisers coincide, any retention difference is path-dependent:
   require matched distance travelled (KL from start, relative update
   norm) or a Pareto frontier over step budgets before crediting P1.
5. If Y is sparsity: require float32 or a threshold sweep, and the
   learning rate and number of passes of each arm. If either differs
   by more than a factor of two, report the comparison as step-confounded.
6. If Y is retention and P3 differs between arms: test target tracking
   with prefixes held fixed (frozen vs moving-average self-teacher on
   the same trajectories, same step budget) before crediting P1 or P3.
   In fine-tuning, regenerating the stored response moves P1 with P3;
   credit such a contrast to both.
7. If Y is transfer to harder or longer instances: P1 is the expected
   cause (imitation-learning support argument); check that the gain
   survives a step-matched off-policy arm and a later RL stage.
8. Output: the property or bundle credited, and the comparison that
   would separate the remaining candidates.
```

Table 3. Predicted outcomes of four accounts under five manipulations (predictions derived by the author from each account). "=" means no difference predicted between the arms.

Manipulation	A1 prefix source	A2 target tracking	A3 step budget	A4 precision
On- vs off-policy distillation, fixed teacher, matched step	on retains more	=	=	=
Self-SFT once vs per epoch, matched step	per-epoch retains more	per-epoch retains more	=	=
GRPO vs SFT at matched KL travelled	GRPO retains more	GRPO retains more	=	=
Same run, bfloat16 vs float32	=	=	=	sparsity only in bf16
Harder-variant transfer, matched step	on-policy better	=	=	=

The first row is what Piskorz et al. (2026) ran, with the step matched by learning rate within one loss family, and its outcome, no retention difference, counts against A1 for retention and is predicted by A2 and A3. The second row is close to what Chen et al. (2025) ran, without a formal step match, and its outcome, less forgetting with per-epoch samples, is predicted by both A1 and A2, so it does not separate them. The fourth row is what Shenfeld et al. (2025) report, and favours A4 for sparsity. The fifth row's outcome in the distillation study favours A1 for transfer. The third row is the one that would separate A1 and A2 from A3 for the fine-tuning versus reinforcement learning gap itself, and the author found no study that runs it with the step matched by distance travelled rather than by nominal learning rate.

11. The Strongest Case Against This Paper

The strongest objection is that the distinction between prefix source and target tracking is artificial in practice. In reinforcement learning, the two are the same act: one samples from the current model and reinforces what was sampled, so there is no procedure that has one without the other. A critic could say that "on-policy data" was always shorthand for this joint act, and that a study which separates them in distillation tests a property nobody claimed. The answer is that the distillation study itself, and the on-policy distillation literature it addresses, do treat prefix source as the operative property: on-policy distillation keeps a fixed external teacher and changes only who writes the trajectories, and its weight-space behaviour has been read as a signature of on-policy post-training on that basis (Yu et al., 2026). If the joint act is what protects, that promotion is mistaken in the same way the broad reading of the null is. Either way, the distinction does work.

A second objection is that the minimiser argument is a capacity-limit argument and the runs are short. A 1-billion-parameter student trained for 150 steps is nowhere near the teacher, so the shared destination is never reached and the trajectory source might still steer the compromise. This is correct, and Section 4 says so: the argument predicts that the trajectory source acts through support and path, not that it has no effect. The prediction that its effect on retention should be small relative to the step budget is still a prediction, and it is the one the data bear out. A study with much longer training, or with students closer in capacity to their teachers, could find a larger trajectory effect on retention, and that would count against this paper's reading.

A third objection is that the target-tracking evidence is itself confounded. Iterative-SFT differs from Self-SFT not only in regenerating its targets but in how many times each stored sequence is seen, and RL's Razor's oracle experiment is on a toy task. Both points are fair. Regeneration and reuse are hard to separate, because a target that is regenerated is by construction not reused; the clean test would hold the number of passes per sequence fixed while varying whether sequences come from the initial or the current model, which Section 13 specifies. The oracle result's scope is a single small task and should not be read as more than an existence proof that a nearby target suffices.

A fourth objection is that this paper relies on cross-study comparisons in Figure 1, mixing models, thresholds and training recipes. Panel (a) does mix checkpoints from different organisations and recipes, and the learning rates come from a second study's audit of their documentation. The panel is offered to show that the on-policy label does not order those points, which survives the mixing, not to estimate a dose-response curve, which would not.

12. What Is Not Known

It is not known whether trajectory source affects retention when the target tracks the model. No published comparison holds a tracking target fixed in form, as SDFT's self-teacher or a reweighted self-distribution, while switching only between student-sampled and teacher-sampled prefixes.

It is not known whether reinforcement learning's retention advantage over fine-tuning survives a comparison matched on distance travelled from the start. The available matched comparisons hold the nominal learning rate fixed or compare Pareto frontiers; none fixes the KL from the starting policy and then compares forgetting.

It is not known how much of the Self-SFT versus Iterative-SFT gap is target tracking and how much is the number of passes over each stored sequence.

It is not known whether update sparsity of reinforcement-learning checkpoints differs from that of fine-tuning checkpoints at matched learning rate and number of passes in float32. The controlled evidence covers distillation; the reinforcement-learning evidence is either in bfloat16 on released checkpoints or reports the absence of sparsity in float32 without a fine-tuning comparison at matched step.

It is not known whether the distillation study's results hold at scales where the student is close to the teacher, for long reasoning traces beyond a single run, or with teachers from a different family.

It is not known whether the harder-variant advantage of on-policy distillation persists through later training stages in general; in the one case reported, it did not survive reinforcement learning.

13. Studies That Would Settle It

Five comparisons would assign the credit, each changing one property of Section 2.

A target-tracking factorial in distillation. Cross trajectory source (student, teacher) with teacher type (frozen external teacher; frozen self conditioned on a demonstration; exponential-moving-average self conditioned on a demonstration). Match the step budget by KL travelled on target inputs. The target-tracking account predicts that retention improves with tracking and does not depend on trajectory source; the prefix-source account predicts a trajectory effect in every teacher condition.

A regeneration-versus-reuse test in fine-tuning. Fine-tune on self-generated correct samples under four conditions: generated once from the initial model and seen once; generated once and seen for several epochs; regenerated each epoch and seen once; regenerated each epoch with the same total passes as the second condition. This separates whether retention comes from a tracking target or from not repeating stored sequences.

A distance-matched fine-tuning versus reinforcement learning comparison. Train both until each reaches the same KL from the start on target inputs, at several values, and compare forgetting at each value. If reinforcement learning still forgets less at equal distance, something beyond distance travelled is at work, and the distance account is incomplete; if not, the advantage reduces to reinforcement learning's tendency to travel less for a given gain, which is the RL's Razor claim in its sharpest form.

A precision-controlled sparsity comparison. Repeat matched fine-tuning, distillation and reinforcement-learning runs in float32 and bfloat16 at the same learning rates and passes, and report sparsity at a sweep of thresholds and relative to the update norm. The precision account predicts that differences vanish in float32 at matched step.

A support-targeted transfer test. Train on-policy and off-policy distillation arms at matched step on a task family with a difficulty parameter, and evaluate transfer as a function of how far the student's own trajectories at test time fall outside the teacher's training trajectories. The support account predicts that the on-policy advantage grows with that distance and is near zero when the student stays on teacher-like trajectories.

14. Conclusion

The controlled distillation study answers the question it set: in distillation toward a fixed teacher, whose trajectories the loss is computed on has little effect on forgetting or update sparsity, while the learning rate has a large one. That answer is predicted by the structure of

the comparison, in which both arms pursue the same destination. It does not reach the claim that reinforcement learning retains because its training target is rebuilt from the model as the model changes, because neither distillation arm has such a target, and the strongest retention evidence on that side, from self-generated targets that are regenerated against ones that are not, varies that property together with prefix source. The sparsity claim was already in doubt for reasons of precision and step size, and the on-policy label does not order the checkpoints it was built on. The one effect the distillation study attributes to trajectory source, transfer to harder instances, is the effect that trajectory source should have. Calling all of this "on-policy" has let one label carry seven properties; separating them shows which of the reported effects belong to which.

References

Agarwal, R., Vieillard, N., Zhou, Y., Stanczyk, P., Ramos, S., Geist, M., & Bachem, O. (2023). On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. arXiv:2306.13649. doi:10.48550/arXiv.2306.13649.

Armandpour, M., Ilhan, F., Harrison, D., Jaiswal, A., Hoang, D. N. M., Faghri, F., Zhang, Y., Cho, M., et al. (2026). Unmasking On-Policy Distillation: Where It Helps, Where It Hurts, and Why. arXiv:2605.10889. doi:10.48550/arXiv.2605.10889.

Bengio, S., Vinyals, O., Jaitly, N., & Shazeer, N. (2015). Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. arXiv:1506.03099. doi:10.48550/arXiv.1506.03099.

Biderman, D., Portes, J., Gonzalez Ortiz, J. J., Paul, M., Greengard, P., Jennings, C., King, D., Havens, S., et al. (2024). LoRA Learns Less and Forgets Less. arXiv:2405.09673. doi:10.48550/arXiv.2405.09673.

Chan, A., Silva, H., Lim, S., Kozuno, T., Mahmood, A. R., & White, M. (2021). Greedification Operators for Policy Optimization: Investigating Forward and Reverse KL Divergences. arXiv:2107.08285. doi:10.48550/arXiv.2107.08285.

Chen, H., Razin, N., Narasimhan, K., & Chen, D. (2025). Retaining by Doing: The Role of On-Policy Data in Mitigating Forgetting. arXiv:2510.18874. doi:10.48550/arXiv.2510.18874.

Chen, Y., Zhu, T., Zhang, M., Chen, X., Huang, J., Xu, D., & Wang, Y. (2026). On-Policy Replay for Continual Supervised Fine-Tuning. arXiv:2605.29495. doi:10.48550/arXiv.2605.29495.

Chu, T., Zhai, Y., Yang, J., Tong, S., Xie, S., Schuurmans, D., Le, Q. V., Levine, S., et al. (2025). SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training. arXiv:2501.17161. doi:10.48550/arXiv.2501.17161.

Di, Q., Li, X., Ji, K., Zhang, C., Zhao, H., & Gu, Q. (2026). Understanding Off- vs On-Policy Distillation: A Tale of Distinct Training Objectives. arXiv:2609.38666. doi:10.48550/arXiv.2609.38666.

Gu, Y., Dong, L., Wei, F., & Huang, M. (2023). MiniLLM: On-Policy Distillation of Large Language Models. arXiv:2306.08543. doi:10.48550/arXiv.2306.08543.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. arXiv:1503.02531. doi:10.48550/arXiv.1503.02531.

Huan, M., Li, Y., Zheng, T., Xu, X., Kim, S., Du, M., Poovendran, R., Neubig, G., et al. (2025). Does Math Reasoning Improve General LLM Capabilities? Understanding Transferability of LLM Reasoning. arXiv:2507.00432. doi:10.48550/arXiv.2507.00432.

Jia, N., Yang, H., Ma, X., Lian, J., Zhang, S., Zhang, W., Zeng, K., Cai, X., et al. (2026). Asymmetric On-Policy Distillation: Bridging Exploitation and Imitation at the Token Level. arXiv:2605.06387. doi:10.48550/arXiv.2605.06387.

Kalajdzievski, D. (2024). Scaling Laws for Forgetting When Fine-Tuning Large Language Models. arXiv:2401.05605. doi:10.48550/arXiv.2401.05605.

Khan, R. M. S., Liu, Z., Tan, Z., Fleming, C., & Chen, T. (2026). TMS: Trajectory-Mixed Supervision for Reward-Free, On-Policy SFT. arXiv:2602.03073. doi:10.48550/arXiv.2602.03073.

Kim, Y., & Rush, A. M. (2016). Sequence-Level Knowledge Distillation. arXiv:1606.07947. doi:10.48550/arXiv.1606.07947.

Koh, W., Lee, H., Lee, K., Park, H., Lee, D., Hwang, D., & Lee, M. (2026). Continually Post-training Small Models via Synthetic Self-distillation. arXiv:2610.13113. doi:10.48550/arXiv.2610.13113.

Korbak, T., Perez, E., & Buckley, C. L. (2022). RL with KL penalties is better viewed as Bayesian inference. arXiv:2205.11275. doi:10.48550/arXiv.2205.11275.

Lai, S., Zhao, H., Feng, R., Ma, C., Liu, W., Zhao, H., Lin, X., Yi, D., et al. (2025). Reinforcement Fine-Tuning Naturally Mitigates Forgetting in Continual Post-Training. arXiv:2507.05386. doi:10.48550/arXiv.2507.05386.

Li, Y., Zuo, Y., He, B., Zhang, J., Xiao, C., Qian, C., Yu, T., Gao, H., et al. (2026). Rethinking On-Policy Distillation of Large Language Models: Phenomenology, Mechanism, and Recipe. arXiv:2604.13016. doi:10.48550/arXiv.2604.13016.

Liang, H., Jin, Y., Balasubramanian, K., & Lai, L. (2026). Learning Style, Forgetting Semantics: A Case Study of SFT and RFT on Classification Tasks. arXiv:2610.02437. doi:10.48550/arXiv.2610.02437.

Lin, Y., Lin, H., Xiong, W., Diao, S., Liu, J., Zhang, J., Pan, R., Wang, H., et al. (2023). Mitigating the Alignment Tax of RLHF. arXiv:2309.06256. doi:10.48550/arXiv.2309.06256.

Lin, X., Sang, H., Wang, Z., & Zhang, X. (2025). Debunk the Myth of SFT Generalization. arXiv:2510.00237. doi:10.48550/arXiv.2510.00237.

Luo, M., Wang, Z., Zhou, Z., Ye, B., Zhao, J., Zhang, M., & Wei, T. (2026). RL Forgets! Towards Continual Policy Optimization. arXiv:2607.04364. doi:10.48550/arXiv.2607.04364.

Matsutani, K., Takashiro, S., Minegishi, G., Kojima, T., Iwasawa, Y., & Matsuo, Y. (2025). RL Squeezes, SFT Expands: A Comparative Study of Reasoning LLMs. arXiv:2509.21128. doi:10.48550/arXiv.2509.21128.

Ming, R., Wu, H., Hu, S., He, Z., & Yu, B. (2025). One-Token Rollout: Guiding Supervised Fine-Tuning of LLMs with Policy Gradient. arXiv:2509.26313. doi:10.48550/arXiv.2509.26313.

Mukherjee, S., Yuan, L., Hakkani-Tur, D., & Peng, H. (2025). Reinforcement Learning Finetunes Small Subnetworks in Large Language Models. arXiv:2505.11711. doi:10.48550/arXiv.2505.11711.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., et al. (2022). Training language models to follow instructions with human feedback. arXiv:2203.02155. doi:10.48550/arXiv.2203.02155.

Piskorz, J., Berthon, A., & van der Schaar, M. (2026). On-Policy or Off-Policy Learning? A Systematic Study of Distillation Dynamics. arXiv:2609.35259. doi:10.48550/arXiv.2609.35259.

Ren, X., Wu, B., & Liu, L. (2024). I Learn Better If You Speak My Language: Understanding the Superior Performance of Fine-Tuning Large Language Models with LLM-Generated Responses. arXiv:2402.11192. doi:10.48550/arXiv.2402.11192.

Rofin, M., Varre, A., & Flammarion, N. (2026). (How) Learning Rates Regulate Catastrophic Overtraining. arXiv:2604.13627. doi:10.48550/arXiv.2604.13627.

Rojas Nunez, J., Sawant, V., Allen, N., Amgalanbaatar, N., Zongo, Y., Sharma, V., & Chaudhary, M. (2026). Mechanistic origins of catastrophic forgetting: why RL preserves circuits better than SFT?. arXiv:2605.28860. doi:10.48550/arXiv.2605.28860.

Ross, S., Gordon, G. J., & Bagnell, J. A. (2010). A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. arXiv:1011.0686. doi:10.48550/arXiv.1011.0686.

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., & Abbeel, P. (2015). Trust Region Policy Optimization. arXiv:1502.05477. doi:10.48550/arXiv.1502.05477.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. arXiv:1707.06347. doi:10.48550/arXiv.1707.06347.

Shenfeld, I., Pari, J., & Agrawal, P. (2025). RL's Razor: Why Online Reinforcement Learning Forgets Less. arXiv:2509.04259. doi:10.48550/arXiv.2509.04259.

Shenfeld, I., Damani, M., Hubotter, J., & Agrawal, P. (2026). Self-Distillation Enables Continual Learning. arXiv:2601.19897. doi:10.48550/arXiv.2601.19897.

Song, M., & Zheng, M. (2026). A Survey of On-Policy Distillation for Large Language Models. arXiv:2604.00626. doi:10.48550/arXiv.2604.00626.

Springer, J. M., Goyal, S., Wen, K., Kumar, T., Yue, X., Malladi, S., Neubig, G., & Raghunathan, A. (2025). Overtrained Language Models Are Harder to Fine-Tune. arXiv:2503.19206. doi:10.48550/arXiv.2503.19206.

Sun, Y., Chen, M., Zhao, T., Miao, P., Zhang, Z., Shen, H., Zhu, R., & Yin, J. (2026). Talking to Yourself: Defying Forgetting in Large Language Models. arXiv:2602.20162. doi:10.48550/arXiv.2602.20162.

Tajwar, F., Singh, A., Sharma, A., Rafailov, R., Schneider, J., Xie, T., Ermon, S., Finn, C., et al. (2024). Preference Fine-Tuning of LLMs Should Leverage Suboptimal, On-Policy Data. arXiv:2404.14367. doi:10.48550/arXiv.2404.14367.

Tang, Y., Guo, D. Z., Zheng, Z., Calandriello, D., Cao, Y., Tarassov, E., Munos, R., Pires, B. A., et al. (2024). Understanding the performance gap between online and offline alignment algorithms. arXiv:2405.08448. doi:10.48550/arXiv.2405.08448.

Tang, Y., & Munos, R. (2025). On a few pitfalls in KL divergence gradient estimation for RL. arXiv:2506.09477. doi:10.48550/arXiv.2506.09477.

Wu, T., Tao, C., Wang, J., Yang, R., Zhao, Z., & Wong, N. (2024). Rethinking Kullback-Leibler Divergence in Knowledge Distillation for Large Language Models. arXiv:2404.02657. doi:10.48550/arXiv.2404.02657.

Wu, C., Tam, Z. R., Lin, C., Chen, Y., Sun, S., & Lee, H. (2025a). Mitigating Forgetting in LLM Fine-Tuning via Low-Perplexity Token Learning. arXiv:2501.14315. doi:10.48550/arXiv.2501.14315.

Wu, Y., Zhou, Y., Ziheng, Z., Peng, Y., Ye, X., Hu, X., Zhu, W., Qi, L., et al. (2025b). On the Generalization of SFT: A Reinforcement Learning Perspective with Reward Rectification. arXiv:2508.05629. doi:10.48550/arXiv.2508.05629.

Wu, C. H., Zhang, T., & Raghunathan, A. (2026). Off-Policy Merging Beats On-Policy Self-Distillation for Continual Learning. arXiv:2610.05872. doi:10.48550/arXiv.2610.05872.

Xu, S., Fu, W., Gao, J., Ye, W., Liu, W., Mei, Z., Wang, G., Yu, C., et al. (2024a). Is DPO Superior to PPO for LLM Alignment? A Comprehensive Study. arXiv:2404.10719. doi:10.48550/arXiv.2404.10719.

Xu, W., Han, R., Wang, Z., Le, L. T., Madeka, D., Li, L., Wang, W. Y., Agarwal, R., et al. (2024b). Speculative Knowledge Distillation: Bridging the Teacher-Student Gap Through Interleaved Sampling. arXiv:2410.11325. doi:10.48550/arXiv.2410.11325.

Yang, Z., Pang, T., Feng, H., Wang, H., Chen, W., Zhu, M., & Liu, Q. (2024). Self-Distillation Bridges Distribution Gap in Language Model Fine-Tuning. arXiv:2402.13669. doi:10.48550/arXiv.2402.13669.

Yu, G., Liu, W., Hu, Y., Ma, H., Jiang, J., & Ye, H. (2026). Dense Supervision, Sparse Updates: On the Sparsity and Geometry of On-Policy Distillation. arXiv:2606.13657. doi:10.48550/arXiv.2606.13657.

Yuan, L., Chen, W., Zhang, Y., Cui, G., Wang, H., You, Z., Ding, N., Liu, Z., et al. (2025). From f(x) and g(x) to f(g(x)): LLMs Learn New Skills in RL by Composing Old Ones. arXiv:2509.25123. doi:10.48550/arXiv.2509.25123.

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Yue, Y., Song, S., & Huang, G. (2025). Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?. arXiv:2504.13837. doi:10.48550/arXiv.2504.13837.

Zhang, D., Xu, Y., Wang, H., Chen, Q., & Peng, H. (2026a). Good SFT Optimizes for SFT, Better SFT Prepares for Reinforcement Learning. arXiv:2602.01058. doi:10.48550/arXiv.2602.01058.

Zhang, M., Liu, Y., Lin, S., Yang, X., Dai, Q., Luo, C., Jiang, W., Hou, P., et al. (2026b). Towards On-Policy SFT: Distribution Discriminant Theory and its Applications in LLM Training. arXiv:2602.12222. doi:10.48550/arXiv.2602.12222.

Zhu, H., Zhang, Z., Huang, H., Su, D., Liu, Z., Zhao, J., Fedorov, I., Pirsiavash, H., et al. (2025a). The Path Not Taken: RLVR Provably Learns Off the Principals. arXiv:2511.08567. doi:10.48550/arXiv.2511.08567.

Zhu, W., Xie, R., Wang, R., Sun, X., Wang, D., & Liu, P. (2025b). Proximal Supervised Fine-Tuning. arXiv:2508.17784. doi:10.48550/arXiv.2508.17784.

Zhu, S., Ye, X., Lu, H., Shi, W., & Liu, G. (2026). The Many Faces of On-Policy Distillation: Pitfalls, Mechanisms, and Fixes. arXiv:2605.11182. doi:10.48550/arXiv.2605.11182.

© 2026 Pranay Mahendrakar. Licensed under CC BY 4.0.

DOI: 10.5281/zenodo.24537485