FORGOTTEN OR INACCESSIBLE? RESTORATION SAVINGS REVEAL WHAT LANGUAGE MODELS RETAIN AFTER FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Catastrophic forgetting is commonly defined as a substantial drop in performance on previously learned tasks after training on new tasks or data. However, this behavioral failure does not establish that the underlying knowledge was erased. We test this distinction with *restoration savings*, which asks whether, given the same small amount of retraining, a forgotten ability is easier to recover than a comparable ability that was never learned. We teach a model a random subset of facts it did not know, induce forgetting by fine-tuning it on an unrelated task, and then retrain learned and never-learned facts under this same budget. Previously learned facts are recovered far more easily, even when they are not themselves retrained and when they are asked in new wordings. Together, these findings show that behavioral access can be lost while a recoverable trace of prior learning remains.

1 INTRODUCTION

Language models are adapted after pre-training through supervised fine-tuning (Ouyang et al., 2022), preference optimization (Rafailov et al., 2023), or reinforcement learning (Shao et al., 2024). This often makes them fail on questions they previously answered correctly (Luo et al., 2025; Wang et al., 2023; Harmon et al., 2026), a phenomenon known as catastrophic forgetting (McCloskey & Cohen, 1989; French, 1999). Such failures are usually read as evidence that the underlying knowledge was erased, yet the correct answer may only have become harder to access (Kotha et al., 2024). Standard evaluations cannot tell these two cases apart (Sun et al., 2025; Thede et al., 2026). Sampling more responses may recover an answer that greedy decoding misses (Reisizadeh et al., 2025; Li et al., 2026), but no finite number of samples can prove that the answer is gone (Hanley & Lippman-Hand, 1983). Retraining does not settle the question either, since with enough training a model can learn a forgotten fact just as it can learn a new one (Hu et al., 2025; Yoon et al., 2026). What separates the two cases is cost: if prior learning survives, a forgotten answer should be easier to recover than one the model never learned, given the same limited training.

We formalize this comparison as *restoration savings*, a test based on relearning. We refer to training on a new task as *interference*, and call an ability *evicted* when the model expressed it before interference but no longer expresses it afterward under a fixed evaluation. We measure expression by the *access rate*, the fraction of questions the model answers correctly under this evaluation. We then retrain, or *restudy*, the evicted ability and a matched control ability that the model never expressed. Both receive comparable data and the same *restudy budget*, a small, fixed number of training steps. Restoration savings is the difference in access rate between the evicted ability and the control after restudy. The test follows recovery tests in machine unlearning (Deeb & Roger, 2024; Siddiqui et al., 2026). A positive gap does not imply that the weights still encode the ability as they did before interference. It shows only that prior learning continues to make relearning cheaper after the model has lost access to the ability.

To make this test causal, we control which facts the model learns (Figure 1). Comparing facts a model already knows with facts it does not is confounded, since known facts may simply be easier. We therefore start from questions the base model answers incorrectly and randomly split them into three sets (step 1). We teach the model two of them, A and C, and withhold the third, B (step 2). We then fine-tune the model on an unrelated task, DBpedia classification (Zhang et al., 2015), which

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.

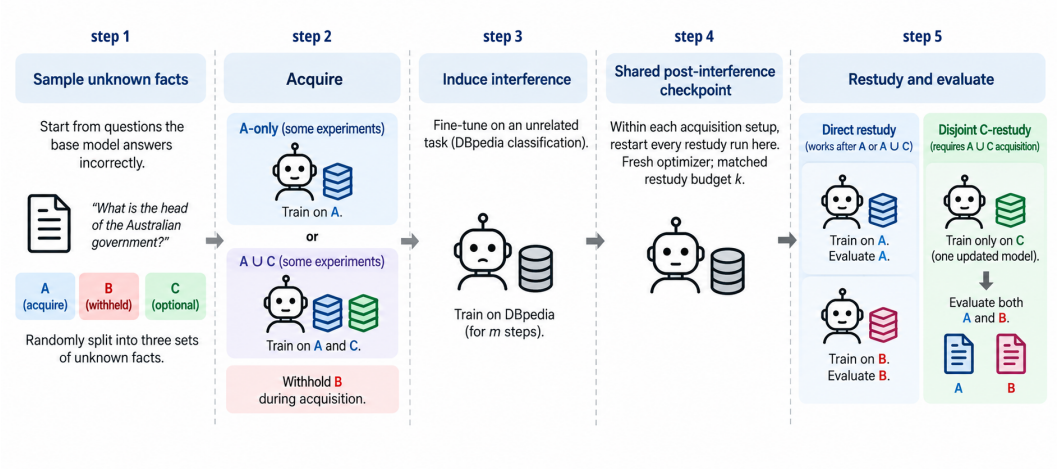

Figure 1: **Randomized restoration-savings protocol.** A and C are acquired; B is withheld. All restudy arms start from the same post-interference checkpoint. Direct restudy trains A or B; disjoint restudy trains only C.

suppresses access to the acquired facts (step 3). From this shared post-interference checkpoint (step 4), we restudy A and B separately for the same small budget and compare how well each recovers (step 5). Because assignment is random, any recovery advantage of A over B is caused by the added learning rather than by differences between the facts. On Llama-3.2-3B, interference drives the access rate on A from 0.94 to 0.04, yet after only four gradient updates of restudy it recovers to 0.85, while B remains at 0.06.

Two further results go beyond showing that forgotten facts can be relearned. First, recovery does not require restudying the forgotten facts themselves. This is what set C is for: it was learned together with A, so in step 5 we can restudy only C and evaluate both A and B on the resulting model. This *disjoint restudy* restores A far more than B, even when the questions are asked in wordings never seen during training. Second, a forgetting score does not reveal how much can still be recovered. The paper makes three contributions:

- We introduce restoration savings, which measures how much more a forgotten ability recovers than a never-learned one under the same training budget, and use random assignment of facts within a single model to show that prior learning causally improves recovery after interference.
- We show that the effect holds when restudying separate facts, under wordings never seen in training, and when an answer counts as correct only if it exactly matches the reference. These controls rule out explanations based on rehearsing the evicted facts, replaying the exact training prompts, or lenient grading of answers.
- We show that forgetting scores miss how much can be recovered: as interference increases, models whose access rate on acquired facts is identically zero can differ by more than a factor of two in what the same small update restores.

2 Restoration Savings

Restoration savings asks whether a model recovers forgotten knowledge more easily than knowledge it never learned, given the same small amount of training. We measure it with the five-step protocol in Figure 1, then define the statistics we report and the variants of the restudy step used in our experiments.

2.1 PROTOCOL

(1) Sample unknown facts. We start from factual questions that the base model answers incorrectly. Selecting items the base model answers incorrectly ensures that, under our evaluation, access rate at the start of acquisition is zero, so any later access is attributable to our fine-tuning rather than to pre-training. We randomly split these questions into three sets of equal size: two sets that will be learned, A and C, and a set B that will be withheld. Random assignment ensures that the sets do not differ systematically, for example in difficulty, so any later difference between them is caused by what we teach.

(2) Acquire. Facts the model learned during pre-training are not suitable for our test, because we do not know how they were learned. We therefore teach the facts ourselves, fine-tuning the model once on the union $A \cup C$ and confirming that it answers the facts in both sets correctly before continuing. Because A and C are learned together, in the same run and with the same settings, they have the same acquisition history. The two sets play different roles later: A is the acquired target that we evaluate, while C is used only as a disjoint restudy set (Section 2.3). The model never sees the facts in B. They are the control: B goes through every later step exactly like A, so the only difference between the two is that A was taught and B was not.

(3) Interfere. Next, we fine-tune the model on a different task, which we call *interference*. This mirrors post-training in practice, where a model is trained on new data unrelated to what it learned before, and any forgetting is a side effect rather than a goal. In our main experiments, interference is DBpedia classification, which produced the sharpest loss of access among the tasks we tried (Section 3). We check that access rate on the acquired sets drops substantially before continuing. We call a fact *evicted* when the model answered it correctly before interference but fails afterward.

(4) Shared checkpoint. Let θ_{int} denote the model parameters after interference (for LoRA runs, the base weights together with the trained adapter). This checkpoint is the common starting point for every comparison. Because all restudy runs begin from the same weights, any difference in how A and B recover cannot come from differences in how the model was trained before restudy.

(5) Restudy and evaluate. Starting from θ_{int}, we briefly retrain the model on a set of facts, which we call *restudy*, and then measure access rate on A and B. We write k for the restudy budget, the number of restudy steps $k{=}8$ (Appendix B). Each restudy run starts from θ_{int} with a fresh optimizer, so runs differ only in which facts are restudied and for how many steps. We keep k small because, with enough training, the model learns any fact regardless of its history (Section 4.2); the question is how much prior learning helps when training is limited.

What eviction does and does not mean. Eviction is defined by a specific test: we ask each question with a fixed prompt, take the single most likely answer (greedy decoding; Holtzman et al., 2019), and check it against the correct answer. A fact that fails this test may still be answered correctly with a different prompt or by sampling several answers. Even if every sample fails, the correct answer may retain a small probability that a limited number of samples misses. Eviction therefore establishes only that the model no longer produces the answer under our test. It says nothing about whether the knowledge was lost or retained. Restoration savings does not settle that question either. It asks something narrower: whether prior learning still makes the fact easier to restore than a fact the model never learned.

2.2 MEASURING RESTORATION SAVINGS

Throughout, the access rate of a model on a set of facts is the fraction of questions whose generated answer passes the verifier described in Section 3. We write $\text{Access}(\theta, S)$ for the access of the model with parameters θ on a set S. Let θ_{int} denote the parameters after interference (step 4), and let $R_k(\theta_{\text{int}}; S)$ denote the parameters obtained by restudying only the set S for k steps, starting from θ_{int} with a fresh optimizer. A step is one microbatch; Appendix B gives the number of examples and optimizer updates per step.

Access rate before and after restudy. The starting access rates are those at the post-interference checkpoint,

$$A_0 = \text{access}(\theta_{\text{int}}, A), \qquad B_0 = \text{access}(\theta_{\text{int}}, B).$$

In *direct restudy*, each set is restudied on its own and evaluated on itself:

$$A_k = \text{access}\big(R_k(\theta_{\text{int}}; A), A\big), \qquad B_k = \text{access}\big(R_k(\theta_{\text{int}}; B), B\big).$$

In *disjoint restudy*, only C is restudied, and both sets are evaluated on the same resulting model:

$$A_k = \text{access}\big(R_k(\theta_{\text{int}}; C), A\big), \qquad B_k = \text{access}\big(R_k(\theta_{\text{int}}; C), B\big).$$

We use the same symbols A_k and B_k for both variants and state the variant with every result.

Final gap. Restoration savings is the difference in access rate between the acquired and withheld sets after the same restudy budget,

$$\mathcal{S}(k) = A_k - B_k. \tag{1}$$

A positive gap means that, for the same amount of training, previously learned facts end up more accessible than facts the model never learned. If interference had completely erased what was learned, A would behave like B and the gap would be near zero.

Gain gap. If some acquired facts survive interference, A starts restudy with a head start ($A_0 > B_0$), and the final gap partly reflects it. To measure only the improvement produced by restudy, we also report the *gain gap*,

$$\mathcal{S}^{\text{gain}}(k) = (A_k - A_0) - (B_k - B_0) = \mathcal{S}(k) - (A_0 - B_0), \tag{2}$$

which removes the starting difference $A_0 - B_0$. The gain gap matters most when a substantial access rate on acquired facts survives interference. It is a descriptive adjustment, not a causal lower bound, since a set that starts higher has less room to improve.

2.3 RESTUDY VARIANTS

We use the protocol in three ways, each answering one of the questions. Each variant rules out one more alternative explanation for why acquired facts might recover better.

Direct restudy (Q1): does prior learning help recovery? We run two separate restudies from θ_{int}: one trains on A and evaluates A, the other trains on B and evaluates B, with the same budget, learning rate, and shuffle seed. Because the facts were assigned at random, A and B are equally hard on average, and the only difference between them is that A was taught before interference. If A recovers more than B under the same restudy, that difference must come from the earlier teaching.

Disjoint restudy (Q2): does recovery require retraining the forgotten facts? The model learned A and C together, and interference suppresses both. If the model relearns only part of what it was taught, does the rest come back too? We restudy only C and then evaluate both A and B on the resulting model. Neither set is trained on, and both are tested on the same model, so any gap between them can only come from A having been taught. A positive gap means that relearning part of what was forgotten restores the rest, which would not happen if each fact had been erased separately. It does not explain what makes this possible.

Untrained wordings (Q3): does recovery depend on the exact question wording? A model could appear to recover a fact only because it relearns the answer to one specific question string, not the fact itself. To test this, we need to control every wording the model sees, so we use fictional facts about invented people. The model learns each fact through two wordings, for example "What is N's profession?" and "N's occupation is". It is restudied through a third, "N works as a", and tested through a fourth, "What does N do for a living?", which never appears in training. As in Q2, we restudy only C. If A still recovers more than B under the fourth wording, recovery is not tied to the prompts used in training.

Pretrained knowledge. Forgetting also affects what a model learned during pre-training, not only facts we taught it. We therefore also run the protocol on such knowledge. Here we cannot teach the facts ourselves, so steps (1) and (2) change. A contains questions the base model already answers correctly. B contains questions it answers incorrectly, with answers of similar length. Interference, restudy, and evaluation then proceed as before. Because we did not choose which facts the model knows, A and B may differ in more than prior learning; for example, known facts may be easier. This comparison therefore tests whether the effect extends to pretrained knowledge, while the randomized protocol establishes the cause.

3 Experimental Setup

Models and training. We study Llama-3.2 (1B/3B), Llama-3.1-8B (Grattafiori et al., 2024), and Qwen3 (1.7B/4B) (Yang et al., 2025). Most experiments use rank-64 LoRA (Hu et al., 2021) on attention and MLP projections; selected 1B and 3B runs update all parameters. Interference is SFT on DBpedia-14 classification (Zhang et al., 2015), which produced the sharpest loss of access among the tasks we tried; GSM8K (Cobbe et al., 2021) and free-form summarization serve as generative boundary tests (Appendix I). Facts are SQuAD-v2 questions (Rajpurkar et al., 2018) asked without their context passage, except in the untrained-wording study, which uses fictional facts. Hyperparameters and per-study budgets are in Appendix B.

Evaluation and uncertainty. Answers are generated by greedy decoding. An answer is correct if the first line of the model's response contains one of the accepted gold answers as a complete word or phrase, after normalization. Unlike exact match, this accepts a correct answer surrounded by extra words; results under exact match are in Appendix C. Conditions being compared share seeds, batching, learning rates, and data order. We chose $k = 8$ as the main budget before running the later experiments and report all budgets. Results are mean $\pm$ sample SD across seeds, with each gap computed within a seed first. The SD shows how much runs vary, not the uncertainty of the mean; bootstrap intervals are in Appendix C.1. Unless stated otherwise, results use Llama-3.2-3B and $k = 8$.

4 Results

We answer the three questions of Section 2.3 in turn (Sections 4.1–4.3), then ask whether equally forgotten models recover equally (Section 4.4) and how general the effect is (Section 4.5).

4.1 Prior learning causes a recovery advantage (Q1)

Facts the model was taught recover far better than facts it was not, even though it no longer answers either. Here, A, B, and C are random splits of 450 SQuAD-v2 questions that the base model answers incorrectly (150 each), and the model is taught $A \cup C$ before interference. Acquisition raises access rate on A to 0.94 and interference lowers it to 0.04, while withheld facts in B stay near zero throughout (Figure 2a). After eight restudy steps, A recovers to 0.85 while B stays at 0.06 (Figure 2b). The resulting gap of 0.79 ± 0.05 across seven seeds is almost entirely produced by restudy, since only 0.04 of it existed beforehand (gain gap 0.75 ± 0.09; Table 1), and it persists under exact-match scoring (0.74 ± 0.06; Appendix C). Because the two sets were assigned at random, the gap is caused by acquisition rather than by any difference between the facts. The advantage is specific to small budgets: with longer restudy, B is learned from scratch and the two curves meet by k=64.

The advantage lies in where the acquired facts start, not in how fast they improve. Before restudy, the model already assigns their correct answers higher probability than those of withheld facts, and this probability advantage does not consistently grow during restudy (Appendix C). Acquired facts are simply closer to being answered correctly, so a small update is enough to bring them back.

4.2 Recovery does not require retraining the forgotten facts (Q2)

Restudying only C, the facts taught together with A, restores A nearly as well as restudying A itself (Figure 3a). Because training on C might improve factual answering in general rather than restore

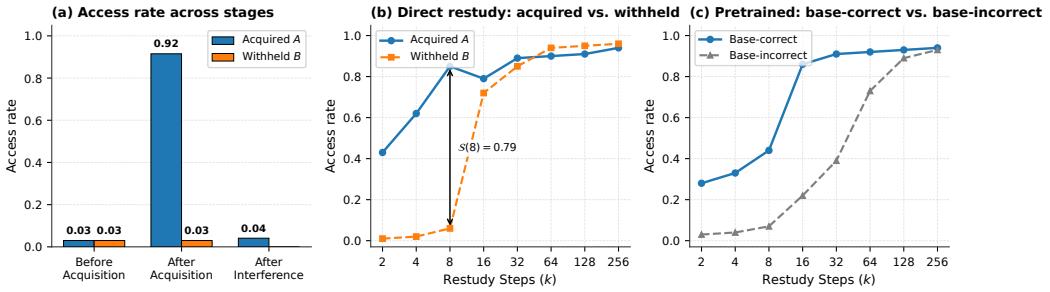

Figure 2: **Prior learning causes a recovery advantage.** (a) Access rate on acquired A and withheld B at each stage. (b) Direct restudy of A and B from the post-interference checkpoint. (c) The same comparison for pretrained knowledge: base-correct vs. base-incorrect facts. Each budget is an independent run.

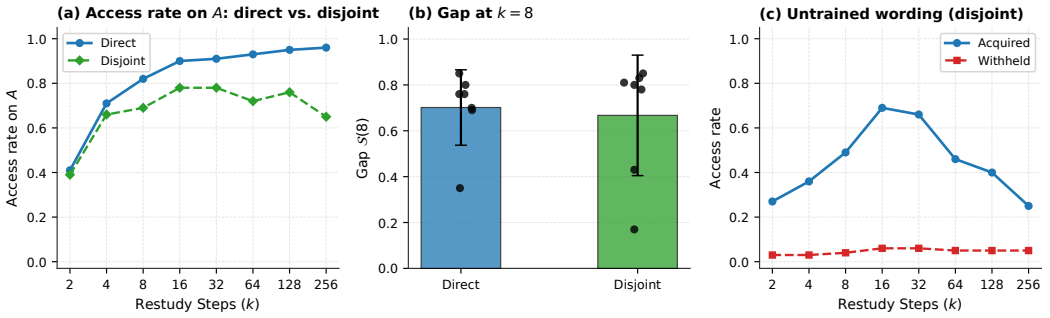

Figure 3: **Recovery does not require retraining the evaluated facts.** (a) access rate on A after direct restudy of A or disjoint restudy of C. (b) Acquired–withheld gap at k=8; dots are seeds, error bars one SD. (c) Disjoint restudy of fictional facts, evaluated under a wording never seen in training. Each budget is an independent run.

what was learned about A, we evaluate B on the same model: B was never taught, so the gap $A - B$ isolates what returns because A was taught. After eight steps, this gap is 0.67 ± 0.26 under disjoint restudy, compared with 0.70 ± 0.17 when A and B are each restudied directly (Figure 3b; separate run from Section 4.1). The difference, 0.04 ± 0.20, has a bootstrap interval reaching 0.19, so direct restudy may still add something. Whichever is the case, restudying C alone raises A far above B even though A received no training. The comparison between direct and disjoint restudy is only informative at small budgets. With enough restudy, direct restudy teaches B from scratch and the direct gap goes to zero (Figure 2b). Under disjoint restudy, B is never trained and stays near zero, so the gap remains large by construction rather than because A recovers better.

The restudied facts do not even need to have been taught. To test this, we restudy a fourth set D of base-incorrect questions that the model never learned, and again evaluate A and B. Restudying D still gives an $A - B$ gap of 0.64, so relearning any factual questions restores much of what was taught. Restudying C gives 0.78, larger by 0.14; averaged over both restudy prompt formats, this restorer-history effect is 0.19 ± 0.07, positive in all five seeds (Appendix D). Facts learned alongside A therefore restore it more. The same pattern holds in a second model family (Section 4.5).

4.3 Recovery transfers to untrained wordings (Q3)

Recovery also holds under a question wording the model never saw in training. Because each SQuAD question has a single wording, recovery there could reflect a memorized prompt–answer mapping rather than the fact itself. We therefore repeat the disjoint design with three sets of 120 fictional person–attribute facts whose wordings we control: each fact is learned through two templates, restudied through a third, and tested through a fourth that appears in no gradient step (Section 2.3;

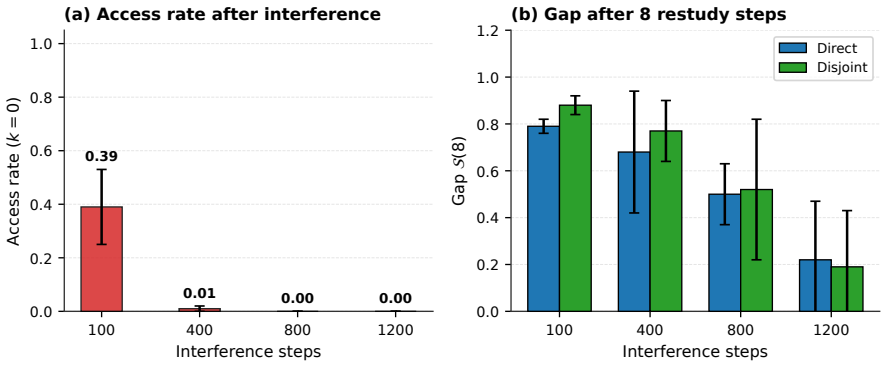

Figure 4: **Equal failure does not imply equal recoverability.** Acquisition is fixed while interference length varies. (a) Access rate on acquired facts after interference (k=0). (b) Acquired–withheld gap after eight restudy steps under direct and disjoint restudy.

Appendix E). Interference lowers access rate on the acquired sets from about 0.78 to about 0.01. After restudying only C, acquired facts lead withheld ones by 0.45 ± 0.26 under the fourth, untrained wording, positive in all seven seeds (Figure 3c). We used this wording when tuning the training setup, so it may favor our recipe. Two more wordings, written afterwards and used only for evaluation, give similar gaps: 0.35 ± 0.22 and 0.43 ± 0.23. Under the untrained wording, access rate on A peaks between 16 and 32 steps and declines with longer restudy, as disjoint access rate of A does more mildly in panel (a); the transfer is therefore strongest at small budgets, where our main comparison is made.

4.4 Equally forgotten models do not recover equally

The access rate after interference, the usual measure of forgetting, does not reveal how much can still be recovered. To show this, we keep acquisition fixed and vary interference over 100, 400, 800, and 1200 steps, with the same five seeds at each level (Figure 4). The access rate on acquired facts is 0.39 after 100 steps, 0.01 after 400, and exactly zero in every seed after 800 and 1200 (Figure 4a). By this measure, the last three conditions are equally forgotten.

Recovery tells them apart. While the access rate after interference is flat at zero from 800 steps onward, the gap after eight restudy steps falls steadily as interference grows: from 0.79 ± 0.03 at 100 steps to 0.22 ± 0.25 at 1200 under direct restudy, and from 0.88 ± 0.04 to 0.19 ± 0.24 under disjoint restudy (Figure 4b). Between 800 and 1200 steps, panel (a) shows no change, yet both gaps drop by more than half, from 0.50 ± 0.13 to 0.22 ± 0.25 (direct) and from 0.52 ± 0.30 to 0.19 ± 0.24 (disjoint). This is the central observation of the paper: equal behavioral failure does not imply equal recoverability.

A smaller gap does not mean the facts are beyond reach; it means they cost more to recover. At 800 steps of interference, doubling the disjoint restudy budget from eight to sixteen steps raises the gap from 0.52 ± 0.30 to 0.78 ± 0.06, close to what eight steps restore after 400 steps of interference. Interference therefore makes recovery more expensive by degrees rather than switching it off.

4.5 The effect generalizes across knowledge sources, models, and tuning modes

Pretrained knowledge. The advantage also appears for facts the model learned during pre-training. Here, A contains 150 SQuAD-v2 questions the base model already answers correctly and B an equal number it answers incorrectly, matched on answer-token length (Section 2.3). Interference lowers access rate on A from 1.00 to 0.01, and even 16 samples per question recover at most 0.05. After eight restudy steps, A leads B by 0.39 ± 0.09 (Figure 2c). Because A and B were not assigned at random, A might be easier for reasons other than being known. We rule out several: the gap remains between 0.40 and 0.55 when B is instead matched on answer likelihood, when question–answer pairs are scrambled, when A keeps only facts the base model answers reliably, and when restudy uses

Table 1: **Recovery advantages at the same small budget.** Final and gain gaps (Equations 1–2) after $k = 8$ restudy steps; mean $\pm$ sample SD of paired seed gaps under the original answer scorer. Randomized, factorial, untrained-wording, and Qwen rows use seven seeds; the observational headline uses nine and its controls three. Llama-3.2-3B unless named.

Comparison	Final gap, k=8	Gain gap, k=8
Randomized acquisition		
Randomized acquisition	0.79 ± 0.05	0.75 ± 0.09
Factorial: direct restudy	0.70 ± 0.17	0.66 ± 0.15
Factorial: disjoint restudy	0.67 ± 0.26	0.62 ± 0.28
Untrained wording, disjoint (development view)	0.45 ± 0.26	0.44 ± 0.25
Qwen3-1.7B factorial, 800 interference steps: direct	0.79 ± 0.04	0.64 ± 0.13
Qwen3-1.7B factorial, 800 interference steps: disjoint	0.80 ± 0.04	0.65 ± 0.13
Observational: base-correct vs. base-incorrect		
Answer-length matched (headline)	0.39 ± 0.09	0.38 ± 0.09
Initial-likelihood matched	0.40 ± 0.18	0.39 ± 0.18
Binding-scrambled control	0.43 ± 0.16	0.42 ± 0.15
Reliably base-correct	0.55 ± 0.05	0.54 ± 0.04
Paraphrased restudy	0.42 ± 0.11	0.41 ± 0.10
Full-parameter tuning	0.33 ± 0.06	0.27 ± 0.06

Table 2: **Restoration savings across models and tuning modes.** Observational comparison under the original answer scorer; mean $\pm$ sample SD over nine seeds for Llama-3.2-3B LoRA and three otherwise. The training cost of $k = 8$ restudy steps differs between setups and is given in Appendix B. †Halved restudy learning rate; one of three seeds has zero gap at eight steps.

Model	Tuning	Access rate drop	$\mathcal{S}(8)$	Gain gap, k=8
Llama-3.2-1B	LoRA	0.998 ± 0.004	0.33 ± 0.06	0.33 ± 0.06
Llama-3.2-3B	LoRA	0.99 ± 0.01	0.39 ± 0.09	0.38 ± 0.09
Llama-3.1-8B†	LoRA	1.00 ± 0.00	0.18 ± 0.17	0.18 ± 0.17
Qwen3-1.7B	LoRA	0.86 ± 0.06	0.48 ± 0.03	0.36 ± 0.08
Qwen3-4B	LoRA	0.90 ± 0.10	0.45 ± 0.02	0.36 ± 0.06
Llama-3.2-1B	Full	0.998 ± 0.004	0.22 ± 0.05	0.21 ± 0.05
Llama-3.2-3B	Full	0.93 ± 0.03	0.33 ± 0.06	0.27 ± 0.06

paraphrased questions (Table 1, bottom; Appendix C.1 explains what each control rules out). One caution remains: what counts as "known" depends on the scorer. Under exact match, only 0.49 of A is answered correctly before interference, because base-model answers often embed the correct span in a longer sentence (Appendix C).

A second model family. The randomized experiment replicates on Qwen3-1.7B: the gain gap is 0.64 ± 0.13 under direct and 0.65 ± 0.13 under disjoint restudy (Table 1; Appendix C.1). Because our main experiments use LoRA, the old answers might survive untouched in the frozen base weights, with restudy merely reconnecting them. We rule this out by applying both interference and restudy to every parameter. The access rate still falls from 1.00 to 0.07, and acquired facts still lead by 0.33 ± 0.06 after both eight and sixteen steps (Appendix H). The observational comparison also extends to two model families from 1B to 8B parameters, and every model shows a positive gap (Table 2, Appendix H). Training settings are not held fixed across sizes, and Llama-3.1-8B required a lower restudy learning rate chosen after the fact. We therefore read this as evidence that the effect is not specific to one model, not as a scaling trend.

Constructed tasks. Synthetic tasks let us control exactly what was learned (Appendix G). In a codebook task, the model learns an arbitrary mapping from classes to answer codes. Interference lowers the greedy access rate on this mapping to zero, and sampling 16 answers recovers at most 0.07. Yet the acquired mapping reaches the access rate threshold after 16 restudy examples, while a matched never-learned mapping needs 64, a four-fold reduction in cost. A model that never learned any code learns at the same rate as the never-learned mapping, and swapping which code is acquired

preserves the advantage. The effect therefore comes from what was learned, not from the codes themselves. Fictional natural-language facts show the same pattern (final gap 0.53 after 64 steps).

5 Related Work

From forgetting to recovery. Classical catastrophic interference studies performance loss during sequential learning (McCloskey & Cohen, 1989; French, 1999), with replay and parameter protection as standard remedies (Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017; Abbes et al., 2026b), and with the choice of adaptation algorithm itself changing how much is lost (Shenfeld et al., 2026; Chen et al., 2025; Jin et al., 2025). Mayo et al. (2023) demonstrate that the model can learn multiple things over time. Our contribution is a randomized factual-acquisition test in language models, with disjoint restudy and explicit measurement controls. In language models, apparent forgetting can also reflect the evaluation protocol or failure to acquire the probed item in the first place (Sun et al., 2025; Harmon et al., 2026). Zheng et al. (2025) specifically attribute spurious forgetting to lost task alignment and show recovery on held-out facts after replaying other facts. Conjugate prompting (Kotha et al., 2024), probabilistic decoding (Reisizadeh et al., 2025), and temporal sampling (Li et al., 2026) can recover outputs hidden from a standard decoder. These methods change how we query or decode the model, without further training. Restoration savings instead changes the weights under a fixed budget and asks whether prior acquisition makes recovery easier.

Relearning and reversibility. Learning-time textual encoding changes later unlearning behavior (Wu et al., 2026), while recent work relates relearning robustness to the alignment of updates on forget and retain data, emphasizing recovery speed rather than eventual recovery (Chen et al., 2026). Closely related LLM work by Deeb & Roger (2024) tests recovery of facts through fine-tuning on disjoint facts. Its random-birthday control also tests a model without prior acquisition of the evaluated facts. Thus disjoint factual recovery and no-acquisition controls are not new to this work. In vision, Siddiqui et al. (2026) show that retain-set fine-tuning restores a trained-and-unlearned model but not a reference model retrained without the forget examples. This also compares acquired and never-acquired states under disjoint recovery.

Our contribution. We apply these relearning tests to forgetting that arises as a side effect of training on another task, rather than from a procedure designed to remove knowledge. Three features distinguish our design (Table 15). First, we randomly assign which facts are acquired within a single model, so the acquired–withheld comparison is causal. Second, we compare acquired and withheld facts at a fixed small budget, since with enough training both are learned. Third, we test disjoint recovery through a question wording absent from every training step. Finally, transport keys (Chaudhury, 2026) test whether a particular representation survives forgetting in vision models, but our result is narrower. It concerns how prior learning affects relearning, not whether any specific representation survives.

Conclusion

Forgetting is usually measured as a drop in performance, and a drop is usually read as erasure. We show that this reading is wrong for language models. Restoration savings, which compares how cheaply forgotten and never-learned knowledge can be relearned, reveals that prior learning survives after the model can no longer produce it. By randomly assigning which facts a model learns, we make this a causal test: taught facts that interference fully suppressed return after few gradient updates, while never-taught facts do not. The retained trace is not tied to specific examples or prompts. Recovery occurs without retraining the forgotten facts, under wordings never seen in training, under exact-match scoring, with full-parameter tuning, and across two model families. Most importantly, standard forgetting scores miss it: models with identical, zero access to acquired facts differ by more than two-fold in what the same small update restores.

Forgetting evaluations should therefore measure not only what a model no longer answers, but what it can still recover. A drop in performance is evidence of lost access, not of lost knowledge. The mechanism behind this retention remains open, and severe forgetting arose here only under classification fine-tuning; we discuss these and other limitations in Appendix A.

AI USE STATEMENT

We used generative AI tools to polish the writing, to help identify related work, and to draft parts of the paper. All AI-drafted text was reviewed and revised by the authors, and every cited work was checked by the authors. AI tools were not used to design experiments or generate data. The authors take full responsibility for all content in this paper.

REFERENCES

Istabrak Abbes, Nizar Islah, Irina Rish, and Sarath Chandar. What does layer-importance reveal about transformers and state-space models? *arXiv preprint arXiv:2609.16537*, 2026a.

Istabrak Abbes, Gopeshh Subbaraj, Matthew Riemer, Nizar Islah, Tsuguchika Tabaru, Hiroaki Kingetsu, Sarath Chandar, and Irina Rish. Revisiting replay and gradient alignment for continual pre-training of large language models. In Sarath Chandar, Razvan Pascanu, Eric Eaton, Bing Liu, Rupam Mahmood, and Amal Rannen-Triki (eds.), *Proceedings of The 4th Conference on Lifelong Learning Agents*, volume 330 of *Proceedings of Machine Learning Research*, pp. 465–486. PMLR, 11–14 Aug 2026b. URL `https://proceedings.mlr.press/v330/abbes26a.html`.

Archie Chaudhury. Forgetting is not erasure: Recovering latent knowledge via transport keys. *arXiv preprint arXiv:2606.02860*, 2026.

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

Yi Chen, Hanna Hsieh, Shuhong Liu, Chuanbo Hua, Zihan Ma, Kun Wang, and Joo-Young Kim. Distance is not enough: Forget-retain alignment gap predicts llm relearning robustness. *arXiv preprint arXiv:2608.25429*, 2026.

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

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

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3 (4):128–135, 1999.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

James A Hanley and Abby Lippman-Hand. If nothing goes wrong, is everything all right?: interpreting zero numerators. *Jama*, 249(13):1743–1745, 1983.

Jackson Harmon, Andreas Hochlehnert, Matthias Bethge, and Ameya Prabhu. Mapping post-training forgetting in language models at scale. In *International Conference on Learning Representations*, volume 2026, pp. 115664–115714, 2026.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

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. *arXiv preprint arXiv:2106.09685*, 2021.

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

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Hangzhan Jin, Sitao Luan, Tianwei Ni, Sicheng Lyu, Guillaume Rabusseau, Reihaneh Rabbany, Doina Precup, and Mohammad Hamdaqa. Rl fine-tuning heals ood forgetting in sft. *arXiv preprint arXiv:2509.12235*, 2025.

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

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

Yuetai Li, Zhangchen Xu, Fengqing Jiang, Bhaskar Ramasubramanian, Luyao Niu, Bill Yuchen Lin, Xiang Yue, and Radha Poovendran. Temporal sampling for forgotten reasoning in llms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28309–28327, 2026.

Jingren Liu, Zhong Ji, YunLong Yu, Jiale Cao, Yanwei Pang, Jungong Han, and Xuelong Li. Parameter-efficient fine-tuning for continual learning: A neural tangent kernel perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *International Conference on Machine Learning*, pp. 23610–23641. PMLR, 2023.

David Mayo, Tyler Scott, Mengye Ren, Gamaleldin Elsayed, Katherine Hermann, Matt Jones, and Michael Mozer. Multitask learning via interleaving: A neural network investigation. In *44th Annual Meeting of the Cognitive Science Society*, 2023.

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

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, 2018.

Hadi Reisizadeh, Jiajun Ruan, Yiwei Chen, Soumyadeep Pal, Sijia Liu, and Mingyi Hong. Leak@ k: Unlearning does not make llms forget under probabilistic decoding. *arXiv preprint arXiv:2511.04934*, 2025.

Yi Ren and Danica Sutherland. Learning dynamics of llm finetuning. In *International Conference on Learning Representations*, volume 2025, pp. 70523–70563, 2025.

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

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

Shoaib Ahmed Siddiqui, Adrian Weller, David Krueger, Gintare Karolina Dziugaite, Michael Curtis Mozer, and Eleni Triantafillou. From dormant to deleted: Tamper-resistant unlearning through weight-space regularization, 2026. URL https://arxiv.org/abs/2505.22310.

Huashan Sun, Yizhe Yang, Yinghao Li, Jiawei Li, and Yang Gao. Unveiling and addressing pseudo forgetting in large language models, 2025. URL https://arxiv.org/abs/2411.11932.

Lukas Thede, Stefan Winzeck, Zeynep Akata, and Jonathan Richard Schwarz. Captrack: Multifaceted evaluation of forgetting in llm post-training. *arXiv preprint arXiv:2603.06610*, 2026.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, et al. Trace: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*, 2023.

Ruihan Wu, Konstantin Garov, and Kamalika Chaudhuri. Learning-time encoding shapes unlearning in llms. In *International Conference on Learning Representations*, volume 2026, pp. 87040–87067, 2026.

Xiaoyu Xu, Xiang Yue, Yang Liu, Qingqing Ye, Huadi Zheng, Peizhao Hu, Minxin Du, and Haibo Hu. Unlearning isn't deletion: Investigating reversibility of machine unlearning in llms, 2026. URL https://arxiv.org/abs/2505.16831.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Sangyeon Yoon, Hyesoo Hong, Wonje Jeung, and Albert No. Rethinking benign relearning: Syntax as the hidden driver of unlearning failures, 2026. URL https://arxiv.org/abs/2602.03379.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

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

A Limitations and Discussion

Scope of the interference regime. Severe forgetting arises in our experiments only under classification SFT. GSM8K and summarization interference leave most factual access intact, so the gaps there are boundary tests under partial forgetting, not replications of the main regime (Appendix I).

What restoration savings establishes. The randomized gap shows that prior acquisition still affects relearning after access is lost. It does not show that an unchanged representation survives. Earlier learning may leave item-specific information, make the parameters easier to retrain (Malladi et al., 2023; Ren & Sutherland, 2025), or help the model return to factual answering more broadly; disjoint restudy does not separate these. Withheld restorers already produce most of the disjoint gap (Section 4.2); previously acquired restorers add a smaller but consistent increment (Appendix D). We also cannot exclude a modest additional benefit of direct rehearsal. Readout diagnostics place the acquired–withheld contrast at the final layer without identifying a mechanism (Appendix F). The evidence is clearer about *whether* acquisition matters than about *how*.

Recovery has a cost. Restudy restores old answers but can lower accuracy on the interfering task, by an amount that depends on the model, the learning rate, and the task (Appendices C and I). Restoration savings is evidence about prior learning, not a recipe for recovery without trade-off.

Wording, scoring, and reproducibility. The untrained-wording facts are synthetic. Neither scorer is a semantic oracle: the span scorer can credit a gold string inside a negated or multi-answer response, and exact match can reject a correct sentence. We do not establish transfer to natural paraphrases of pretrained knowledge, to entailment-style evaluation, or to other languages. Individual high-budget learning curves are sensitive to accelerator nondeterminism (Appendix B.1); our conclusions rest on low budgets and on means across seeds.

B Protocol

All experiments follow the five stages of Section 3. This appendix gives the training recipes, the designs of the factorial and restorer experiments, and how we report uncertainty.

Recipes. Table 3 lists the main studies. All LoRA runs use rank 64, $\alpha = 128$, zero dropout, and adapters on the query, key, value, attention-output, and three MLP projections. Optimization uses AdamW (Loshchilov & Hutter, 2017) with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01, gradient clipping at norm one, a constant learning rate of 3×10^{-4}, and sequences capped at 512 tokens. Randomized sets A, B, and C contain 150 items each.

Table 3: **Training budgets for the main studies**, in steps (one step is one microbatch of four examples; two steps make one optimizer update). Every restudy budget $k \in \{2, 4, \ldots, 256\}$ restarts from the post-interference checkpoint with a fresh optimizer.

Study	Model	Acquisition	Interference	Seeds
Pretrained knowledge	Llama-3.2-3B	—	400	42–44, 46–51
Randomized acquisition	Llama-3.2-3B	300	400	42–44, 46–49
Acquisition-by-restudy factorial	Llama-3.2-3B	600	400	42–44, 46–49
Interference-depth sweep	Llama-3.2-3B	600	100–1200	5 of the above
Untrained wording	Llama-3.2-3B	600	500	1–5, 42, 43
Qwen factorial	Qwen3-1.7B	600	400 / 800	42–44, 46–49

What a step costs. Under the LoRA recipe, k steps present $4k$ examples and perform $k/2$ updates, so the headline budget $k = 8$ is 32 presentations and four updates, not eight passes over a set. The Llama-3.2-3B full-parameter control uses batch size one with accumulation eight ($k = 8$ is one update); the 1B full-parameter control matches the LoRA accounting. Codebook budgets count unique examples. We compare costs only within identical recipes. Budgets are independent: budget k for seed s uses shuffle seed $s + k$.

Choice of headline budget. We fixed $k = 8$ after inspecting the pretrained-knowledge curves, where every budget gives a positive gap (Figure 2c), and before running the factorial and all later experiments. We report every budget.

Scorers and task utility. Access rate uses greedy decoding with a 16-token answer budget (32 in Appendix J). The span scorer of Section 3 is used unless stated; Appendix C adds exact match on the first answer segment and on the whole completion. Two DBpedia scores appear. Calibration gates and the adapter-scaling experiment report accuracy on the first 128 examples of the interference training pool, which measures task fit only. All utility numbers after restudy use 128 held-out examples never seen during interference.

Acquisition-by-restudy factorial. Base-incorrect items are randomly split into A, B, C. The model is trained on $A \cup C$, then interfered once. From this shared checkpoint, each budget runs three arms: restudy A and evaluate A; restudy B and evaluate B; restudy C and evaluate both. With access rates A_d, A_c, B_d, B_c, the interaction $(A_d - A_c) - (B_d - B_c)$ is the direct-minus-disjoint advantage. The two disjoint outcomes come from the same model, so they share exactly the same update. An interval containing zero does not establish equivalence.

Restorer-history split. To test whether the disjoint restorer must itself have been acquired, we split each factorial checkpoint's sets into four disjoint 75-item sets: acquired target A, withheld target B, acquired restorer C, and withheld restorer D (the other half of B). The contrast

$$\left(A_{\text{after } C} - B_{\text{after } C}\right) - \left(A_{\text{after } D} - B_{\text{after } D}\right) \tag{3}$$

evaluates identical targets on both sides, cancelling their starting difference. Appendix D uses the same split.

B.1 Statistics and reproducibility

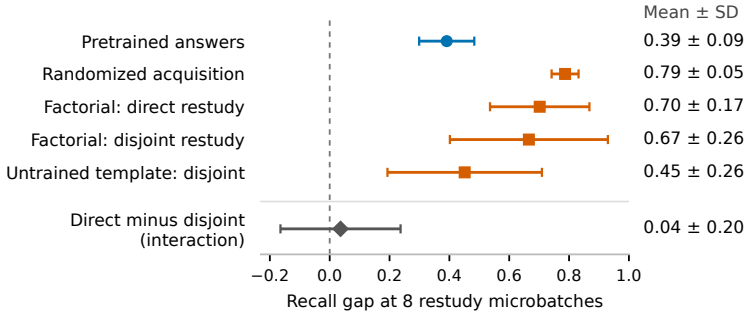

Figure 5: **Low-budget gaps on Llama-3.2-3B** (mean ± one sample SD across seeds). Circles: observational; squares: randomized; diamond: direct-minus-disjoint interaction. The tests use different protocols and are not independent replications.

We report mean ± sample SD across seeds, computing each gap within a seed first. SDs describe run-to-run variation, not uncertainty in the mean (Figure 5). For inference we use percentile bootstraps with 100,000 replicates. When item-level outputs are saved (pretrained and randomized studies), the bootstrap is hierarchical: seeds, then items within seed. Otherwise it resamples seeds. Budget-wise intervals are pointwise and not corrected for multiple comparisons.

Replays of saved checkpoints reproduce mean gaps but not every cell. Under the span scorer, the largest single-cell differences from the original runs are 0.04 (pretrained), 0.03 (randomized), and 0.01 (Qwen factorial). In the Llama factorial, one direct $k = 32$ cell differs by 0.49. We repeated that seed three times from identical initial weights and settings: two L40S runs agreed bitwise, and an A100 run reproduced the original scores. We therefore attribute the discrepancy to nondeterministic kernels (deterministic algorithms were not enabled) and treat high-budget curves as numerically sensitive. The main text reports original runs; Appendix C reports replays.

C Robustness of the Main Result

We replayed all saved post-interference checkpoints from the pretrained, randomized, and factorial studies, saved every completion, and rescored it under all three scorers.

Scoring. The advantage survives exact matching in every study (Table 4). In the Llama factorial, the exact disjoint gap is positive in six of seven seeds; the exception prefixes its answers ("of 1851" for "1851"). Exact gaps are positive on average at every audited budget ($k \leq 32$) for both model families. The scorer matters most for defining what the base model *knows*. The base-correct set scores 0.99 under the span scorer, 0.49 under first-answer match, and zero under whole-completion match, because base completions continue past the answer. We keep the original item sets throughout.

Table 4: **Gaps at $k = 8$ under span and exact scoring.** Replays; mean $\pm$ sample SD across seeds. Exact match is on the first answer segment; whole-completion match differs by at most 0.01.

		Span scorer		Exact match	
Study	Restudy	Final	Gain	Final	Gain
Pretrained knowledge	direct	0.39 ± 0.09	0.38 ± 0.09	0.36 ± 0.09	0.35 ± 0.09
Randomized acquisition	direct	0.79 ± 0.05	0.75 ± 0.09	0.74 ± 0.06	0.70 ± 0.09
Llama factorial	direct	0.71 ± 0.14	0.67 ± 0.13	0.67 ± 0.18	0.63 ± 0.17
	disjoint	0.65 ± 0.28	0.61 ± 0.30	0.58 ± 0.35	0.54 ± 0.38
Qwen factorial (800)	direct	0.79 ± 0.03	0.65 ± 0.13	0.75 ± 0.04	0.61 ± 0.14
	disjoint	0.80 ± 0.04	0.65 ± 0.13	0.72 ± 0.14	0.58 ± 0.14

Retained utility. Whether recovery costs the new task depends on the recipe. In the Llama factorial, held-out DBpedia accuracy falls from 0.92 to 0.84–0.88 at $k = 8$, with large losses in some seeds. In the Qwen factorial it is unchanged (0.96 before, 0.97–0.98 after). Raising the learning rate lowers utility without enlarging the gap (Table 5). One randomized seed never learned DBpedia (accuracy zero throughout). We retain it; its gap (0.68) shows a residual effect of acquisition but not recovery after successful new-task learning.

Table 5: **Learning-rate sensitivity.** Disjoint restudy at $k = 8$ with a fixed data order; exact match; mean $\pm$ sample SD over seven seeds.

Model	LR scale	Final gap	Gain gap	DBpedia accuracy
Llama-3.2-3B	$0.5\times$	0.62 ± 0.25	0.58 ± 0.21	0.96 ± 0.02
	$1\times$	0.67 ± 0.26	0.62 ± 0.24	0.85 ± 0.13
	$2\times$	0.59 ± 0.28	0.55 ± 0.32	0.68 ± 0.32
Qwen3-1.7B	$0.5\times$	0.64 ± 0.08	0.50 ± 0.07	0.97 ± 0.02
	$1\times$	0.72 ± 0.05	0.58 ± 0.16	0.98 ± 0.01
	$2\times$	0.62 ± 0.08	0.48 ± 0.12	0.91 ± 0.04

Likelihood. Acquired answers begin restudy with a large likelihood advantage, and restudy does not consistently widen it. Measured as mean-token log-probability of the primary gold answer, the post-interference gap is 2.10 nats in the pretrained study and 3.71 in the Llama factorial. The $k = 8$ gain gaps are 0.10 ± 0.54 and -0.13 ± 0.48 (direct) and 0.50 ± 1.27 (disjoint). Qwen is the exception: its disjoint gain gap is 1.51 ± 0.54, positive in every seed. Restoration savings is thus a finite-budget behavioral advantage from a head start, not a generally faster optimization rate (Section 3).

Starting gaps. The gain gap subtracts the post-interference gap $A_0 - B_0$. This is below 0.01 in the observational studies, about 0.04 in the Llama randomized and factorial studies, 0.06 for full-parameter Llama-3.2-3B, 0.10 for fictional facts, and 0.15 for the Qwen factorial with 800 interference steps. Withheld access starts at or below 0.011 everywhere except fictional facts (0.06).

C.1 Matching controls, replication, and intervals

Controls for the observational comparison. Base-correct facts might recover better for reasons unrelated to prior learning. Each control in Table 1 targets one such reason. *Answer-length matching* (headline) removes the advantage of short answers. *Initial-likelihood matching* matches the base model's probability of the answer string under a neutral prefix (acquired -6.56, matched withheld -6.66, length-matched withheld -7.95). *Binding scrambling* tests whether restudy relearns frequent strings rather than question–answer links; it is imperfect because contradictory answers may make the withheld task harder. *Reliably base-correct* keeps facts answered correctly in most of 16 samples, excluding lucky greedy hits. *Paraphrased restudy* tests exact-prompt relearning; 35% of rewrites repeat the source question. Two limits apply to all controls: removing SQuAD passages leaves some questions underspecified, and even the randomized test measures learned question–answer associations, not necessarily self-contained knowledge.

Intervals for the headline tests. The nine pretrained seed gaps at $k = 8$ range from 0.22 to 0.52 and are all positive. In the Llama factorial, the direct and disjoint advantages are 0.70 ± 0.17 and 0.67 ± 0.26; their interaction is 0.04 ± 0.20 with 95% interval $[-0.08, 0.19]$, the basis for the non-equivalence caution in Section 4.2.

Qwen replication. With 400 interference steps, Qwen3-1.7B retains substantial access (0.45 on A), so we fixed a calibration rule in advance: on seed 42, choose the smallest DBpedia budget leaving acquired access at most 0.10 with DBpedia accuracy at least 0.90. 800 steps passed; 1200 did not, because it left more access on C. With 800 steps, post-interference access is $0.15/0.16/0.01$ on $A/C/B$, varying from 0.02 to 0.34 across seeds. At $k = 8$ the direct and disjoint gain gaps are 0.64 ± 0.13 and 0.65 ± 0.13 (interaction -0.005 ± 0.021); excluding the calibration seed gives 0.62 and 0.63.

D What Restores Access

These tests reuse the Llama-3.2-3B factorial checkpoints and the 75-item $A/B/C/D$ split of Appendix B. No new acquisition or interference is run unless stated.

Prompting alone does not restore access. Without training, no prompting condition recovers A. Greedy access is 0.04 ± 0.07, rewordings give 0.02–0.03, four-shot prompting gives at most 0.01 (with generic exemplars or exemplars from C or D), and best-of-16 sampling reaches 0.08. B stays at zero. If interference had only disrupted the answer format, showing the format in context should reopen access. It does not, so the loss is in the weights.

Shrinking the adapter restores acquired facts. The LoRA adapter holds both the acquisition and the interference update. Scaling the whole update by s with no data or training (Table 6) raises access on A monotonically to 0.48 at $s = 0.25$ while B stays at or below 0.05. DBpedia accuracy also *rises*, from 0.91 to 0.97–0.98. The interference update therefore overshoots along the adapter's directions, and shrinking it re-exposes the acquisition directions it dominated. Selectivity between A and B is expected, since B never entered the adapter. The intervention is specific to LoRA and to an easy task (the base model already scores 0.81); Appendix H shows savings without it.

Table 6: **Adapter scaling.** Access rate and DBpedia (training-pool) accuracy after multiplying the LoRA update by s; no data or training; seven seeds.

s	Access on A	Access on B	DBpedia accuracy
1.00	0.04 ± 0.07	0.00 ± 0.00	0.91 ± 0.09
0.75	0.11 ± 0.14	0.00 ± 0.01	0.97 ± 0.03
0.50	0.31 ± 0.23	0.01 ± 0.02	0.98 ± 0.02
0.25	0.48 ± 0.07	0.05 ± 0.03	0.97 ± 0.03
0.00	0.00 ± 0.00	0.00 ± 0.00	0.81 ± 0.08

Restudy needs a learnable update, not only QA format. The withheld-restorer result could suggest that any QA-format gradient step reopens access. It does not (Table 7). Restudying D with true answers gives a gap of 0.55 at $k = 8$. The same questions with answers rotated (every binding false) give 0.11, and pseudo-word questions and answers give 0.06. The gap tracks how well each restorer is itself learned, so this rules out format alone without showing that factual content is required. A learnable but content-free restorer matched on restorer access would settle it.

Table 7: **Content ladder.** $A - B$ gap after restudying 75 items of each set, and each set's own access at $k = 32$; seven seeds, canonical QA format.

Restudy set	$k = 2$	$k = 8$	$k = 32$	Restorer access, $k = 32$
True D (true answers)	0.27	0.55	0.66	0.63
Deranged D (false answers)	0.13	0.11	0.22	0.17
Pseudo-word questions/answers	0.07	0.06	0.03	0.03

Previously acquired restorers help more. A first follow-up on the factorial checkpoints gave an acquired-restorer gap of 0.69 ± 0.26 and a withheld-restorer gap of 0.55 ± 0.28, but their difference (Equation 3, 0.15 ± 0.26, median 0.07) was dominated by one seed. We therefore ran a prospective, counterbalanced experiment. For five new Llama-3.2-3B seeds, one of two matched quartets of item sets is acquired (600 steps) and the assignment is swapped in a second run; each seed's outcome averages both. Restorers are acquired or withheld, in canonical or suffix-instruction format, and no arm trains on the evaluated targets. The history effect is positive in all five seeds at both budgets (Table 8). The history-by-format interaction is too variable to show invariance to format. One of the ten runs missed the prespecified post-interference gate (acquired-target access 0.107) and is retained, as the frozen plan specified.

Table 8: **Counterbalanced restorer history by format.** Exact-match $A - B$ target gap; mean $\pm$ sample SD over five seeds.

Restorer	$k = 8$	$k = 32$
Acquired, canonical	0.78 ± 0.02	0.80 ± 0.04
Acquired, suffix-instruction	0.76 ± 0.06	0.81 ± 0.03
Withheld, canonical	0.64 ± 0.04	0.66 ± 0.05
Withheld, suffix-instruction	0.53 ± 0.10	0.67 ± 0.07
History effect	0.19 ± 0.08	0.14 ± 0.04
Format effect	-0.06 ± 0.05	0.01 ± 0.01
History $\times$ format	0.09 ± 0.07	0.01 ± 0.06

Restorer syntax. Restudying C through six prompt wrappers (canonical, bare, instruction, suffix instruction, XML, JSON) while evaluating A canonically gives gaps of 0.69–0.75 at $k = 8$ for every wrapper (Table 9). Only the most edit-distant wrapper differs from canonical, and only at $k = 2$ (-0.07 ± 0.07). Pre-restudy gradient cosine to canonical C does not consistently predict recovery (Spearman 0.89, 0.09, -0.03, 0.66 at $k = 2, 8, 32, 128$). The average hides one failure: in one seed the JSON restorer reaches 0.91 access in JSON format while canonical access on both C and A stays at zero, despite a gradient cosine of 0.84. The model can learn answers in a new format without recovering them in the old one.

E Untrained Wordings

Each seed has three disjoint sets of 30 fictional people with four attributes (profession, home city, instrument, pet) drawn from 16-value vocabularies, giving 120 facts per set. Facts are acquired through two wordings, restudied through a third, and evaluated through a fourth that never appears in training (Table 10). No exact prompt is shared between training and evaluation in any run.

Calibration. On seed 42, without running any restudy arm, we required acquisition access of at least 0.75 on A and C under the evaluation wording, at most 0.20 on B, post-interference access at most

Table 9: **Restorer syntax.** Seven seeds, $k = 8$. d_{tok}: normalized token edit distance to canonical; cos: gradient cosine to canonical C (means).

Format	d_{tok}	$\cos(g_f, g_C)$	$A - B$ gap
Canonical	0.00	1.00	0.69 ± 0.26
Bare	0.32	0.92	0.75 ± 0.21
Instruction	0.50	0.89	0.75 ± 0.21
Suffix instruction	0.55	0.82	0.74 ± 0.25
XML	0.50	0.88	0.74 ± 0.22
JSON	0.52	0.85	0.74 ± 0.22

Table 10: **Wordings for person N.** Both acquisition wordings are used in training; restoration is used for all restudy arms; evaluation only for scoring.

Attribute	Acquisition 1	Acquisition 2	Restoration	Evaluation
Profession	What is N's profession?	N's occupation is	N works as a	What does N do for a living?
Home city	In which city does N live?	N resides in	N's home city is	Where does N make their home?
Instrument	Which instrument does N play?	N's musical instrument is	N is known to play the	What musical instrument is associated with N?
Pet	What kind of pet does N have?	N's pet is a	N keeps a pet	Which animal is N's companion?

0.15, and zero prompt overlap. One acquisition wording narrowly failed; two wordings with 600 steps passed. Only then did we launch all seven seeds, all of which are retained. Because the evaluation wording helped select this recipe, we call it *untrained* rather than untouched, and we added two fresh wordings, fixed before any replay and used only for evaluation: a record view (`Person: N / Requested record field: occupation / Value:`) and an interview view (`Tell me the job held by N. Give only the job name. Reply:`).

Results. Across seven seeds, acquisition raises access under the evaluation wording to $0.78/0.78/0.07$ on $A/C/B$; interference lowers it to $0.01/0.01/0.00$ (best-of-8: $0.04/0.05/0.00$). At $k = 8$, disjoint restudy gives a gap of 0.45 ± 0.26, positive in all seven seeds (range 0.06–0.84); direct restudy gives 0.50 ± 0.17, and their interaction is 0.05 ± 0.25. Both fresh wordings preserve the mean effect (Table 11), though access under them was not measured before interference, so they show post-restudy transfer rather than a full learning–loss–recovery sequence. Access on A peaks at $k = 16$–32 and declines with longer restudy (Figure 6). Gold-answer likelihood gain gaps at $k = 8$ are 1.4–1.6 nats in all three wordings, positive in six of seven seeds. Held-out DBpedia accuracy is 0.89 before restudy and 0.90 after disjoint restudy.

Table 11: **Gaps under the untrained and two fresh wordings.** Exact-match final gaps; mean $\pm$ sample SD over seven seeds; every budget restarts from the post-interference checkpoint.

	Untrained		Record (fresh)		Interview (fresh)	
k	Direct	Disjoint	Direct	Disjoint	Direct	Disjoint
2	0.20 ± 0.12	0.24 ± 0.20	0.05 ± 0.04	0.08 ± 0.12	0.14 ± 0.09	0.19 ± 0.20
4	0.36 ± 0.19	0.33 ± 0.21	0.23 ± 0.18	0.17 ± 0.17	0.30 ± 0.16	0.28 ± 0.19
8	0.50 ± 0.17	0.45 ± 0.26	0.33 ± 0.08	0.35 ± 0.22	0.46 ± 0.15	0.43 ± 0.23
16	0.55 ± 0.26	0.63 ± 0.10	0.38 ± 0.13	0.48 ± 0.12	0.52 ± 0.21	0.59 ± 0.12
32	0.67 ± 0.07	0.60 ± 0.10	0.63 ± 0.07	0.48 ± 0.11	0.74 ± 0.05	0.57 ± 0.12

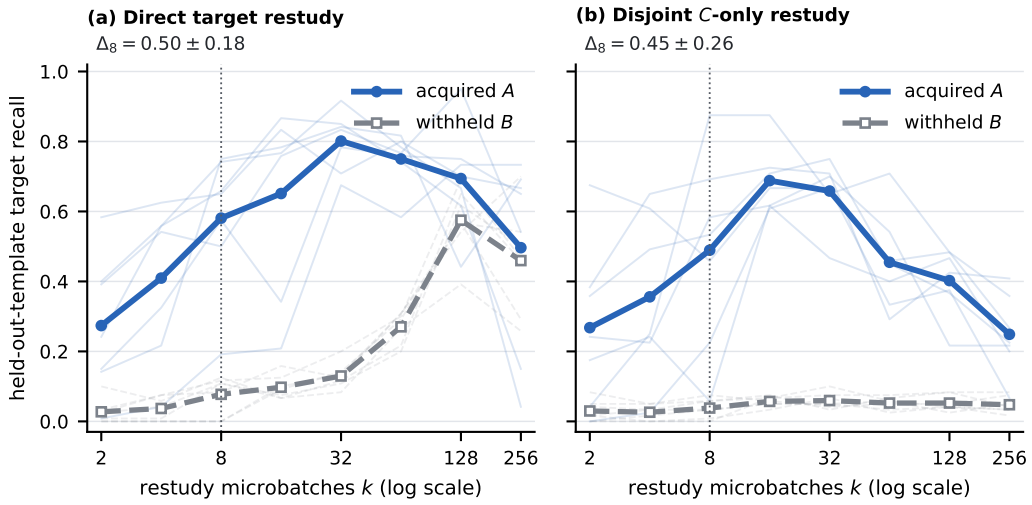

Figure 6: **Restoration under an untrained wording.** Thin lines: seeds; thick lines: means. (a) Direct restudy. (b) Restudying only C restores A while B stays near zero. Annotations: final gap at $k = 8$.

F Readout Diagnostics

On the fictional-fact checkpoints of Appendix J (six seeds per model), we asked whether the recovery advantage is visible in gradients or intermediate layers before restudy (Table 12).

Gradient alignment. Using SFT-loss gradients over all 120 restoration-wording examples per set, the restorer gradient g_C aligns more with acquired than withheld targets after acquisition. After interference the difference nearly vanishes for Llama and becomes variable for Qwen. Gradient alignment, central to a kernel view of fine-tuning (Jacot et al., 2018; Liu et al., 2026), is therefore not a clean predictor of the advantage at this checkpoint.

Logit lens. Applying the model's final normalization and unembedding to the residual stream at five depths (no fitted probe), the acquired-minus-withheld gold-answer likelihood contrast is large only at the final layer, and stays positive after interference in both families. This localizes the clearest measured separation to the readout. It does not show where a fact is stored, and small intermediate contrasts do not show absence. Decomposing where acquisition, interference, and restudy updates land, in the sense of layer necessity versus plasticity (Abbes et al., 2026a), would test whether the readout contrast reflects where these updates write.

Table 12: **Pre-restudy diagnostics.** Gradient column: $\cos(g_C, g_A) - \cos(g_C, g_B)$, computed per seed. Logit-lens columns: acquired-minus-withheld mean-token gold log-probability. Mean $\pm$ sample SD over six seeds; — not measured.

		Gradient	Logit-lens contrast		
Model	Checkpoint	alignment	Block 0	Middle	Final
Llama-3.2-3B	after acquisition	0.18 ± 0.22	0.01 ± 0.41	0.28 ± 0.18	4.33 ± 1.82
	after 400 interf.	—	0.12 ± 0.47	0.19 ± 0.24	1.69 ± 1.15
	after 800 interf.	0.02 ± 0.04	0.06 ± 0.42	0.11 ± 0.31	1.07 ± 0.81
Qwen3-1.7B	after acquisition	0.26 ± 0.28	0.23 ± 0.66	0.27 ± 0.63	6.01 ± 1.12
	after 400 interf.	—	0.28 ± 0.53	0.24 ± 0.76	1.97 ± 1.15
	after 800 interf.	0.15 ± 0.17	0.31 ± 0.66	0.21 ± 0.72	1.99 ± 1.35

G Constructed Restoration Controls

Isomorphic codebooks. Three branches share the same DBpedia inputs. The acquired branch learns a target code, the withheld branch learns a different code with matched label count and token lengths, and a naive branch learns no code. All three then learn the same competing answer format, which lowers greedy access to the acquired code from about 0.95 to zero (best-of-16: at most 0.07). In all three seeds, the acquired branch reaches its criterion after 16 unique restudy examples and the withheld branch after 64, a four-fold cost reduction (Figure 7). The naive branch learns at the withheld rate, so the slowdown is not interference from the earlier code, and swapping which code is acquired preserves the advantage. At 128 examples both branches exceed 0.95: neither failed sampling nor eventual recovery would distinguish them; the matched low-budget comparison does.

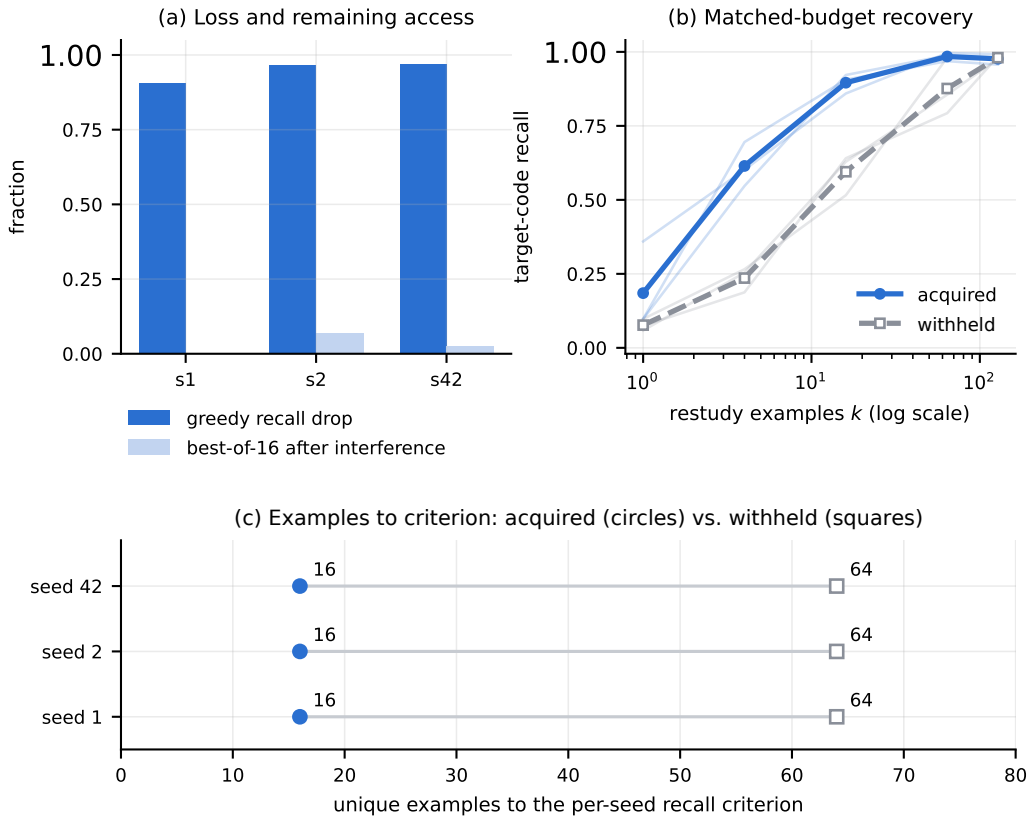

Figure 7: **Restoration savings in an isomorphic codebook.** (a) Interference removes access to the acquired code. (b) Matched restudy curves. (c) Examples to the per-seed criterion $\tau = \min(0.8, 0.8\,\mathrm{Acc}_{\text{post-A}})$.

Fictional natural-language facts. With closed-book facts about fictional entities, interference lowers greedy access on the acquired set by 0.75 (to 0.16, versus 0.06 withheld). The acquired branch reaches 0.65 at 64 restudy steps and 0.89 at 256, versus 0.11 and 0.34 (gain gaps 0.43 and 0.45). Best-of-16 still recovers about half the acquired facts before restudy, so this is savings after greedy access declines rather than after complete failure. Appendix E strengthens this control with randomized sets and an untrained wording.

H Full-Parameter Restoration

LoRA leaves the base weights frozen, so recovery might rely on untouched parameters. Updating all parameters during interference and restudy, Llama-3.2-3B access falls from 1.00 to 0.07 (best-of-16:

Table 13: **Model comparisons at two budgets** (observational design). Nine seeds for Llama-3.2-3B LoRA, three otherwise. †Restudy learning rate halved after the shared rate plateaued at high budgets; low-budget evidence only.

Model	Tuning	Access drop	$\mathcal{S}(8)$	$\mathcal{S}(16)$	Gain gap, $k = 8$
Llama-3.2-1B	LoRA	1.00 ± 0.00	0.33 ± 0.06	0.36 ± 0.11	0.33 ± 0.06
Llama-3.2-3B	LoRA	0.99 ± 0.01	0.39 ± 0.09	0.41 ± 0.08	0.38 ± 0.09
Llama-3.1-8B†	LoRA	1.00 ± 0.00	0.18 ± 0.17	0.30 ± 0.26	0.18 ± 0.17
Qwen3-1.7B	LoRA	0.86 ± 0.06	0.48 ± 0.03	0.45 ± 0.06	0.36 ± 0.08
Qwen3-4B	LoRA	0.90 ± 0.10	0.45 ± 0.02	0.47 ± 0.05	0.36 ± 0.06
Llama-3.2-1B	Full	1.00 ± 0.00	0.22 ± 0.05	0.23 ± 0.05	0.21 ± 0.05
Llama-3.2-3B	Full	0.93 ± 0.03	0.33 ± 0.06	0.33 ± 0.04	0.27 ± 0.06

0.17–0.29), and acquired facts still lead by 0.33 at both $k = 8$ and 16 (gain gap 0.27). The 1B gaps are 0.22 and 0.23 with residual access below 0.01 (Table 13). Savings therefore does not require frozen base weights, though training need not have changed every parameter that encodes the old answers.

A single-seed randomized full-parameter pilot on Llama-3.2-1B (learning rate 5×10^{-5}) lowers access on A/C from $0.93/0.87$ to $0.25/0.27$. At $k = 8$, exact-match gain gaps are 0.49 for both direct and disjoint restudy. DBpedia accuracy falls from 0.97 to 0.84 (direct) and 0.90 (disjoint).

I Generative Interference

Neither generative task passed its prespecified severe-forgetting gate. Both are therefore boundary tests under partial forgetting.

Summarization. With 1600 steps of Dolly-15k summarization (the precommitted boundary after no development depth passed), access on A is unchanged (0.78 before and after interference), versus 0.05 on B. At $k = 8$, the exact-match gain gaps are small (0.12 ± 0.11 disjoint, 0.06 ± 0.15 direct), and restudy lowers summarization quality (ROUGE-L -0.13 and -0.17).

GSM8K. A development seed with 400 interference steps failed the gate (access on A/C fell only from $0.90/0.89$ to $0.77/0.82$). A prespecified depth search selected 800 steps (access $0.56/0.61$); 1600 steps left *more* access ($0.67/0.66$), so forgetting is not monotonic in training length. On five fresh seeds, the 800-step recipe did not generalize: access on A falls only from 0.86 to 0.70, and only one seed meets every criterion.

Most of the final gap therefore predates restudy. At $k = 8$ the span-scorer final gaps are 0.79 (direct) and 0.86 (disjoint), but the starting gap is already 0.69, leaving gain gaps of 0.10 ± 0.07 and 0.18 ± 0.08, positive in every seed. Disjoint gains stay positive in every seed at every budget; direct gains turn negative by $k = 16$. Exact matching shrinks the starting gap (0.32 first-answer, 0.07 whole-completion) more than the final gap, so its gains are larger. This shows scoring dependence, not a rescued generality test. Recovery is costly: GSM8K accuracy falls from 0.20 to 0.04–0.05 at $k = 8$ in every seed and arm, and no tested learning rate avoids this.

J Prospective Prediction of Disjoint Recovery

Design. Does a short restoration probe predict later recovery beyond baseline correctness and gold-answer likelihood? We froze the design before observing results. Llama-3.2-3B and Qwen3-1.7B each use seeds 70–72 for fitting and 73–75 for testing, with 120 fictional facts in each of A, B, C and interference of 0, 400, or 800 DBpedia steps. A two-step probe restudies half of C; an independent eight-step outcome restudies the other half, both from the same post-interference checkpoint. Separate ridge-regularized logistic predictors per model family use pre-restudy correctness and gold log-probability (baseline), plus probe correctness and the probe's likelihood change (augmented). A second comparison gives both predictors the acquisition label. We compare held-out Brier scores; uncertainty is the SD across the three test seeds.

Result. The probe helps on average but not consistently (Table 14). Mean Brier scores improve from 0.155 to 0.142 (Llama) and from 0.155 to 0.115 (Qwen), but the per-seed improvements are -0.03, -0.02, 0.09 for Llama and 0.01, 0.00, 0.11 for Qwen: one test seed drives the mean. Conclusions are unchanged under the span scorer. A short probe sometimes adds information, but not reliably enough to justify its cost over zero-update likelihood.

Table 14: **Brier-score reduction from adding the probe.** Positive values favor the probe; mean $\pm$ SD across three held-out seeds.

Model	Subset	Baseline	With acquisition label
Llama-3.2-3B	all (primary)	0.013 ± 0.071	0.007 ± 0.064
	positive depth	0.054 ± 0.076	0.046 ± 0.067
Qwen3-1.7B	all (primary)	0.040 ± 0.058	0.012 ± 0.025
	positive depth	0.053 ± 0.086	0.016 ± 0.037

K Comparison with Related Recovery Tests

Table 15: **Closest recovery tests.** A no-acquisition control did not acquire the evaluated target. An untrained view evaluates through a wording absent from both acquisition and recovery. Entries summarize each study's primary experiments.

Study	Setting	No-acquisition control	Recovery data	Untrained view
Mayo et al. (2023)	neural-network task switching	relearning comparison	interleaved task practice	—
Zheng et al. (2025)	LLM continual learning	—	held-out-target recovery via task replay	—
Deeb & Roger (2024)	LLM unlearning	yes (App. E)	disjoint, low-leakage facts	—
Hu et al. (2025)	LLM unlearning	—	benign/related	—
Xu et al. (2026)	LLM unlearning	—	target/retain/unrelated	—
Yoon et al. (2026)	LLM unlearning	—	syntactically related	—
Siddiqui et al. (2026)	vision unlearning	retrained without targets	disjoint retain set	—
This work	LLM incidental forgetting	randomly withheld targets	direct or disjoint	yes

L

Example outputs before and after interference The following examples are drawn from the *four-arm mechanism dataset*, rather than from the SQuAD-v2 experiments used for the headline results.

For the factual query

> **Person:** Henrietta Grimsby
> **Requested record field:** instrument played
> **Value:**

the gold answer is `oboe`. Before interference, the model answers correctly:

> **Before interference:** `oboe`

whereas after DBpedia interference it produces:

> **After interference:** `Artist`

The gold-answer log-likelihood also decreases substantially, from approximately -0.53 before interference to -4.99 afterward. Thus, interference not only changes the greedy output, but also makes the correct answer substantially less preferred by the model.

A second query for the same fictional person asks for a different attribute:

> **Person:** Henrietta Grimsby
> **Requested record field:** occupation
> **Value:**

with gold answer `plumber`. In this case, the model produces:

> **Before interference:** `Field`
> **After interference:** `Field`

These examples illustrate a characteristic output-format failure in this dataset. After DBpedia interference, the model often emits generic classification-style labels such as `Artist`, `Field`, or `Company` instead of the requested factual value.