# UNLEARNING CAN HURT WHEN THERE IS NOTHING TO FORGET

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Unlearning is meant to remove one piece of knowledge from a language model and leave the rest intact. Its side effects are measured by how much the model's scores on other topics drop, and the whole drop is charged to the removal. We show that for two widely used methods, RMU and NPO, most of that drop does not come from removing that knowledge. We fine-tune two copies of a model the same way, one on the documents to be forgotten and one on unrelated documents, and give both the identical unlearning run. If the damage came from removing what the model learned, the copy that never learned it would be spared. Under RMU it is not: across three datasets and fictitious authors, five models from four families, 14 fine-tuning seeds and four measures of damage, it loses about as much as the copy that did, including on the closely related topics where damage is largest, and unlearning statements a model never saw costs as much as unlearning what it knows from pretraining. At least 74% of RMU's collateral damage on MMLU, and over 90% on the other two datasets, occurs without the model having learned what was removed, and NPO on MMLU, at the same amount of damage, behaves like RMU. The control detects dependence where it exists: under GradDiff and SimNPO, a copy that learned other documents about the forget topic is damaged less than the taught copy, about half as much under GradDiff, and damage aimed at chosen topics, by gradient ascent or by RMU itself, is detected in nearly every draw. Unlearning evaluations should report collateral damage next to the damage the same run does to a model that never learned the forget documents.

## 1 INTRODUCTION

Unlearning methods for large language models aim to remove specific knowledge, such as a memorised fact, a copyrighted passage or a hazardous capability, while leaving the rest of the model intact (Jang et al., 2023; Eldan & Russinovich, 2023; Li et al., 2024). Benchmarks judge them on two counts: whether the target is gone, and how much else is lost (Maini et al., 2024; Shi et al., 2025; Jin et al., 2024). The second, collateral damage, is measured by scoring held-out knowledge before and after unlearning, and the drop is read as the price of removing the target. RWKU builds a set of neighbouring knowledge around each target to measure this price (Jin et al., 2024), and WMDP reads a drop on closely related fields as a lack of precision (Li et al., 2024).

This reading has a gap. The model after unlearning differs from the model before in two ways: it has lost the target, and its weights have been moved by an optimisation run aimed at the target's documents. Either could lower its score on other topics, and a before-and-after comparison cannot tell them apart. It leaves open the question that decides how to read a collateral-damage number: would the same damage appear if the model had never learned the target?

We answer it with a simple control (Figure 1), in the standard setting where the knowledge to be removed is first taught by fine-tuning, as in TOFU (Maini et al., 2024). Two copies of one model are fine-tuned with the same recipe: the *taught copy* on false statements about the forget topic, the *untaught copy* on false statements about an unrelated topic. Because the base model rarely holds these statements, whether a copy has learned them can be checked directly. Both copies then receive the identical unlearning run, aimed at the forget topic's documents. If the damage comes from removing what those documents taught, the untaught copy, which never learned them, should be



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.

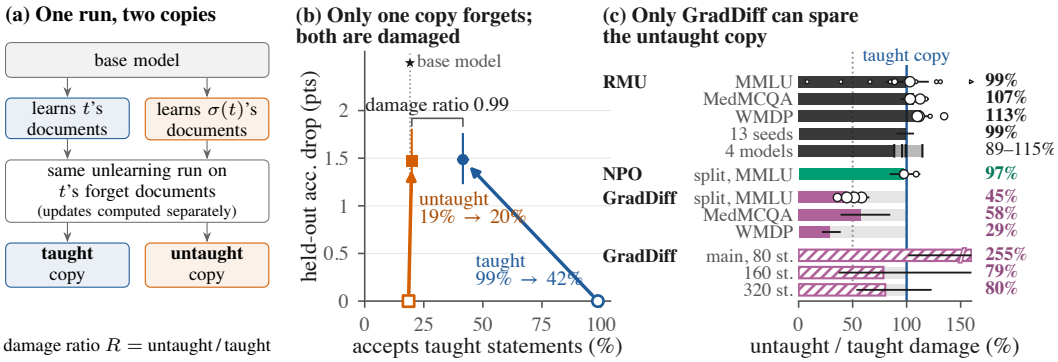


**Figure 1: Under RMU a model never trained on the forget documents loses as much as the model that learned them, although only the latter forgets; under GradDiff, when it learned other documents about the forget topic, it is spared about half.** (a) The two copies differ only in what they learned. (b) RMU on MMLU (Llama-3.1-8B-Instruct, 224 runs): acceptance of the taught statements before (hollow) and after (filled) unlearning, and held-out accuracy drop. (c) Untaught copy's held-out log-probability damage as a share of the taught copy's. RMU bars pool all strengths (seeds 1–13 pooled); NPO: 320 steps; GradDiff split: 160 steps; dots: single strengths (area $\propto$ taught damage); split: the untaught copy learned the other half of $t$'s documents (Section 5.4); hatched: main design; cut at 160%. Intervals: 95% over forget topics.

largely spared. If it is damaged just as much, the damage does not need the target, and the before-and-after number is not the cost of removing it.

For RMU (Li et al., 2024), the untaught copy is not spared. On Llama-3.1-8B-Instruct, the ratio of its held-out accuracy loss to the taught copy's is 0.99 on MMLU topics, 1.04 on MedMCQA and 1.11 on WMDP. It stays near one over 13 further fine-tuning seeds (0.98), on four further models from three other families (0.95 to 1.19), on TOFU's fictitious authors (1.09), under a second pairing of topics (1.07) and on the contrastive score and generated answers, which ignore a shift shared by all options (0.9 to 1.2), and unlearning statements the base model never saw costs as much as unlearning what it knows from pretraining (1.04). On MMLU the taught copy loses 1.49 accuracy points, and its extra damage over the untaught copy is at most 0.38 points, the upper end of the 95% interval: at least 74% of the damage, and over 90% on MedMCQA and WMDP, does not depend on the model having learned the forget documents. Unlearning removes the taught statements only from the taught copy, yet both copies lose about the same everywhere else, even on the topics closest to the target, where damage is largest.

The same control returns a much lower ratio when the damage does depend on what the model learned. Under GradDiff (Liu et al., 2022; Maini et al., 2024), the untaught copy takes about half of the taught copy's damage when it has learned other statements about the forget topic (log-probability ratios 0.29 to 0.58 on three datasets, all with intervals below one), and SimNPO (Fan et al., 2025a) likewise damages it less, while under NPO (Zhang et al., 2024) on MMLU, at the same amount of damage, it takes as much (0.97). And when we aim damage at randomly chosen topics, the held-out measurement picks them out: under gradient ascent they lose more log-probability in 263 of 272 draws, and under RMU itself more accuracy in 46 of 48. How much collateral damage depends on having learned the target is a property of the method, and the control measures it.

Our contributions are:

- **The untaught-copy control** (Section 3): a paired design that measures how much of a collateral-damage number depends on the model having learned what is removed. In a study that unlearns many topics it needs no extra fine-tuning, only one extra unlearning run per forget topic and strength.
- **Evidence that under RMU collateral damage depends little on having learned the target** (Sections 5.1 and 5.2), across three datasets, five models from four families, 14 fine-tuning seeds, TOFU, four measures of damage, knowledge from pretraining and the topics nearest the target.
- **Evidence that the control detects dependence when it exists** (Sections 5.3 and 5.4), from positive controls with gradient ascent and with RMU, and from GradDiff and SimNPO,

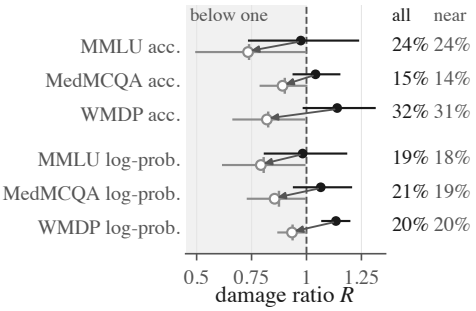


**Figure 2: A dependence on the learned target of 15 to 32% of the damage would move the ratio's interval below one.** Observed $R$ (filled; RMU, Llama-3.1-8B-Instruct, $\sigma(t)$ left out) and $R$ after planting the smallest detected dependence (hollow): the share of the taught copy's damage in the right-hand columns is removed from the untaught copy, spread over all held-out topics (hollow circle, 95% interval) or only over topics in $t$'s domain (tick); Appendix C.4.

which spares an untaught copy that learned other documents about the forget topic, where NPO at the same damage does not.

We recommend that unlearning evaluations report, for every method and strength, the taught copy's damage, the untaught copy's damage, their difference and their ratio.

## 2 Related work

**Evaluating unlearning.** Benchmarks for LLM unlearning measure forgetting on a target set and utility on retained or held-out knowledge: TOFU on fictitious authors (Maini et al., 2024), MUSE on news and books (Shi et al., 2025), WMDP on hazardous knowledge (Li et al., 2024), RWKU on real-world entities (Jin et al., 2024), and OpenUnlearning across many methods and metrics (Dorna et al., 2025). Several papers question what these measurements show: whether the forgotten knowledge can be recovered (Lynch et al., 2024; Hu et al., 2025a; Łucki et al., 2025; Deeb & Roger, 2024; Doshi & Stickland, 2024; Che et al., 2025), whether disjoint forget and retain sets understate damage (Hu et al., 2025b; Jeung et al., 2026), whether benchmark results predict behaviour elsewhere (Thaker et al., 2025; Feng et al., 2025), and whether unlearning matches the goals set for it in policy (Cooper et al., 2025). Recent work studies collateral damage in more detail, auditing which knowledge is at risk before unlearning (Su et al., 2026), measuring how damage concentrates near the target (Dolgova & Stich, 2026), probing for knowledge lost beyond it (Ko et al., 2025), and showing that entangled representations raise it (Wybitul et al., 2026). All of these measure damage against the model before unlearning or a retrained model. Our control asks whether that damage depends on the model having learned what is removed.

**Models that never saw the data.** A model never trained on the forget set usually appears in unlearning evaluation as the goal: TOFU's retain model is trained without the forget set, and unlearning is judged by how close it comes to that model (Maini et al., 2024); it is never itself unlearned; filtering pretraining data builds such models at scale (O'Brien et al., 2026). A few works do run unlearning on models that never saw the forget data, and adversarial requests to unlearn data an image classifier never trained on can destroy its accuracy (Huang et al., 2025). SALMUBench (Selvas-Sala et al., 2026) applies unlearning baselines to a CLIP model trained without its sensitive set, as a sanity check, and finds that some methods still degrade general utility. Kim et al. (2026) add questions the model never learned to one LLM's forget set and find that, under gradient ascent, they harm retained knowledge more than learned ones. Our control pairs two LLM copies that differ only in whether they learned the forget documents, holds the unlearning run fixed, and measures the difference in their damage with uncertainty. It turns the sanity check into a measurement that any collateral-damage number can be read against.

**Controls that remove the cause.** Other fields check an effect by running the procedure where the proposed cause is absent: sham surgery in clinical trials (Moseley et al., 2002), exposures that cannot cause the outcome in epidemiology (Lipsitch et al., 2010), randomised models for saliency maps (Adebayo et al., 2018), and control tasks for probes (Hewitt & Liang, 2019). In machine learning, such checks have already changed the reading of results on emergent abilities (Schaeffer et al., 2023) and ImageNet progress (Recht et al., 2019), and randomised baselines now do the same for interpretability metrics (Heap et al., 2026; Korznikov et al., 2026). The untaught copy applies the same idea to collateral damage. Appendix J discusses further work.

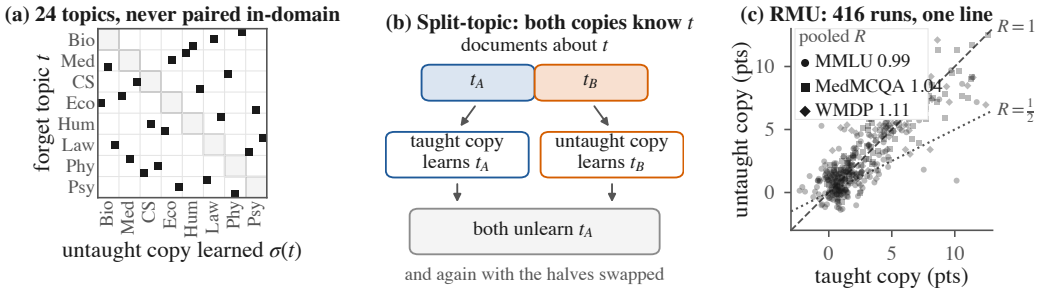


**Figure 3: The control needs no extra fine-tuning, and under RMU every run lies near the line of equal damage.** (a) The pairing $\sigma$ on MMLU: one mark per forget topic $t$ at the topic whose fine-tuned model is its untaught copy. No topic is paired inside its domain (grey blocks), and each fine-tuned model is the taught copy for its own topic and the untaught copy for one other, so a study adds one unlearning run per topic and strength. (b) The split-topic design (Section 5.4). (c) Held-out accuracy drop of both copies in every RMU run on Llama-3.1-8B-Instruct; pooled $R$: ratio of summed damages.

## 3 The untaught-copy control

A collateral-damage number is the change in a model's score on held-out topics from before to after unlearning. We want to know how much of that change depends on the model having learned what is removed.

**Two copies.** Let $\theta_0$ be a base model and $t$ a forget topic. Fine-tuning $\mathcal{A}(\theta_0; t)$ teaches the model a set of documents about $t$, and unlearning $\mathcal{U}(\cdot\,; t)$ then targets the same documents, the forget documents of $t$. The *taught copy* is the model a standard study builds:

$$\theta_t^{\text{taught}} = \mathcal{U}\big(\mathcal{A}(\theta_0; t);\ t\big).$$

The *untaught copy* is fine-tuned with the same recipe on documents about a different topic $\sigma(t)$ and then receives the same unlearning run, still on the forget documents of $t$:

$$\theta_t^{\text{untaught}} = \mathcal{U}\big(\mathcal{A}(\theta_0; \sigma(t));\ t\big).$$

The pairing $\sigma$, fixed before any model was trained, matches each topic with one from a different domain (a group of related topics such as law or clinical medicine; Section 4). Since $\sigma$ covers every topic, each fine-tuned model is the taught copy for its own topic and the untaught copy for another. A full study therefore needs no extra fine-tuning, only one extra unlearning run per forget topic and strength.

**What is held fixed.** The two copies share the base model and the fine-tuning recipe (steps, batch size and learning rate, on up to 50 documents per topic). They receive the same unlearning method at the same strength and schedule, applied to byte-identical forget documents. They differ only in which topic they learned before unlearning. The method computes each copy's update from that copy's own weights, so any difference in the update caused by having learned the documents is part of what we measure.

**Damage and the damage ratio.** We score each copy on the held-out topics $\mathcal{H}_t$, which exclude $t$, just before and just after unlearning. Its damage $D$ is the score before minus the score after, so each copy is compared with itself, as in a standard study. Over a set of runs $r$ (one forget topic at one unlearning strength), the *damage ratio* is

$$R \;=\; \frac{\sum_r D_r^{\text{untaught}}}{\sum_r D_r^{\text{taught}}}. \tag{1}$$

We also report the paired difference $\Delta = \overline{D^{\text{taught}} - D^{\text{untaught}}}$, the extra damage of the taught copy in the measure's own units. $R$ is not bounded by one, and because it divides sums, runs with more damage weigh more.

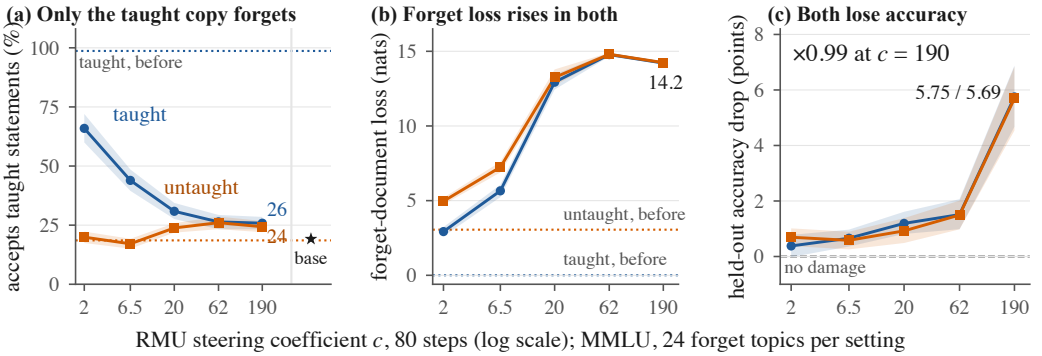


**Figure 4: As RMU grows stronger, only the taught copy forgets the taught statements, while both copies' forget-document loss rises and both lose the same held-out accuracy.** Endpoint dose-response on MMLU: Llama-3.1-8B-Instruct, 80 steps at five steering coefficients $c$, 24 forget topics each, with 95% intervals over topics. (a) Share of taught statements each copy accepts; dotted, each copy before unlearning; star, the base model. (b) Loss on the forget documents. (c) Held-out accuracy drop; ×, untaught over taught at $c = 190$.

**What the comparison measures.** Because the copies differ only in what they learned, $\Delta$ is the damage that having learned the forget documents adds to the same unlearning run, and $\Delta/\overline{D^{\text{taught}}}$ is the share of the reported damage that depends on it. A ratio well below one means the damage depends on having learned the forget documents (Figure 3); $R$ near one and $\Delta$ near zero mean it arises whether or not the model learned them, and a dependence of 15 to 32% of the damage would already push $R$'s interval below one (Figure 2).

**Measures.** We report two measures on the held-out questions. *Accuracy* is the fraction of four-option questions on which the correct option receives the highest log-probability. The *correct-answer log-probability* is the length-normalised log-probability of the correct option; we fixed it as the primary measure for the main experiment before reading the results. We lead with accuracy, which is easier to interpret and depends only on the order of the four options (Section 5.5). Intervals are 95% percentile intervals from a paired bootstrap over forget topics, which resamples all runs of a topic together for both copies. Appendix C defines two further measures, generated-answer accuracy and a contrastive score unchanged when all four options fall together.

## 4 Experimental setup

We follow the usual design for evaluating unlearning of fine-tuned knowledge, as in TOFU (Maini et al., 2024): the model first learns the content to be removed, and unlearning is judged on that content and on everything else. The one change is that every unlearning run is applied to two copies. Appendix Table 1 summarises the experiments and Appendix B gives every detail.

**Topics and documents.** We use 24 topics from each of three multiple-choice datasets: MMLU (Hendrycks et al., 2021), MedMCQA (Pal et al., 2022) and WMDP (Li et al., 2024). The topics fall into domains of related topics: eight on MMLU (such as law, computing and clinical medicine), eight on MedMCQA and three on WMDP (biology, chemistry, cybersecurity). For each topic we build up to 50 fine-tuning documents from its own questions, each stating a question with a wrong option as its answer; documents that repeat an evaluation question are dropped. Because the statements are false, the base model rarely holds them, and learning them is easy to verify: a model accepts a taught statement when it prefers the false option over the other three. The base model accepts 19% of them, the taught copy 99%. Held-out damage is measured on 60 other questions from each of the remaining 23 topics.

**Models and training.** The main model is Llama-3.1-8B-Instruct (Grattafiori et al., 2024). We repeat the MMLU experiment on Qwen2.5-7B-Instruct (Yang et al., 2024), Qwen3-4B-Instruct-2507 (Yang et al., 2025), Granite-3.3-8B-Instruct (IBM Granite Team, 2024) and Zephyr-7B-$\beta$ (Tun-

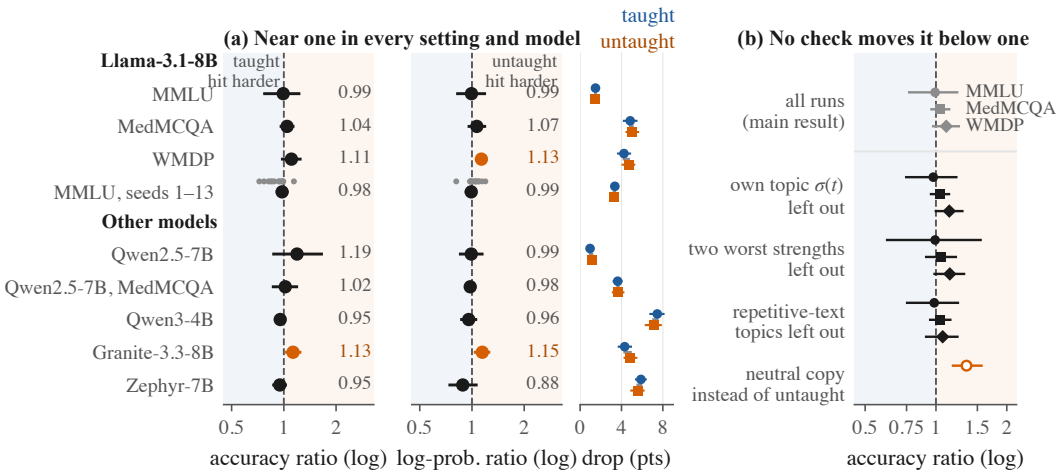


**Figure 5: The pooled damage ratio is near one on every dataset and model, and no robustness check moves it below one.** Untaught over taught held-out damage under RMU, with 95% intervals over forget topics; blue shading marks where the taught copy is hit harder, vermillion where the untaught copy is, and vermillion marks an interval above one. (a) Every setting, in accuracy and correct-answer log-probability; grey dots are the 13 further fine-tuning seeds; right, each copy's mean accuracy drop with its interval. (b) Llama-3.1-8B-Instruct accuracy ratio after leaving out the untaught copy's own topic $\sigma(t)$, the two most damaging strengths or the held-out topics with repetitive text, or with a neutral copy (hollow, MMLU only) in place of the untaught copy; grey, the main result.

stall et al., 2024). Each copy is fine-tuned with LoRA (Hu et al., 2022) for 200 steps, with one fine-tuning seed per dataset; the Llama MMLU experiment is repeated with 13 further seeds.

**Unlearning.** The main method is RMU (Li et al., 2024), which pushes the model's internal activations on the forget documents toward a random direction while keeping them unchanged on a small retain set. The forget documents of $t$ are the statements the taught copy learned; the retain set comes from eight MMLU subjects outside the 24 topics. The unlearning strength varies the number of steps and RMU's steering coefficient $c$: 12 strengths on MMLU (8 on all 24 topics, 4 on 8 of them) and 4 each on MedMCQA and WMDP, for 224, 96 and 96 runs per copy. Section 5.4 and Appendix H report GradDiff, NPO and SimNPO. Leaving out held-out topics on which a run produced repetitive text (1 to 11% of scores) changes no ratio by more than about 0.05 (Appendix D).

## 5 Results

### 5.1 The untaught copy is damaged about as much

Under RMU, the copy that never learned the forget documents loses about as much held-out accuracy as the copy that did. Figure 4 follows the runs through both copies of Llama-3.1-8B-Instruct. Unlearning removes the taught statements from the taught copy, whose acceptance falls from 99% to 42% on MMLU and to about 60% on MedMCQA and WMDP over all runs, and on MMLU to 26% at the strongest coefficient (panel a). The untaught copy accepts 19% before and 20% after, the rate of the base model: it had learned nothing from $t$'s documents. The same run still raises the loss on the forget documents of both copies alike, to 14.2 nats per token at the strongest coefficient on MMLU (panel b). On held-out topics the two copies lose about the same at every strength (panel c), and run by run their damage lies along the line of equal damage on all three datasets (Figure 3). In no row of Figure 5 is the untaught copy damaged significantly less than the taught copy, and in one, WMDP in log-probability, it is damaged significantly more.

Figure 5 gives the damage ratio and both copies' damage for every setting; on Llama the ratio is 0.99, 1.04 and 1.11 in accuracy on MMLU, MedMCQA and WMDP. The paired difference, the taught copy's extra damage (Appendix D), is 0.0, $-0.2$ and $-0.4$ accuracy points, with 95% intervals whose upper ends are 0.38, 0.28 and 0.17 points. At least 74% of the damage on MMLU, 94% on MedMCQA and 95% on WMDP therefore does not depend on the model having learned the forget

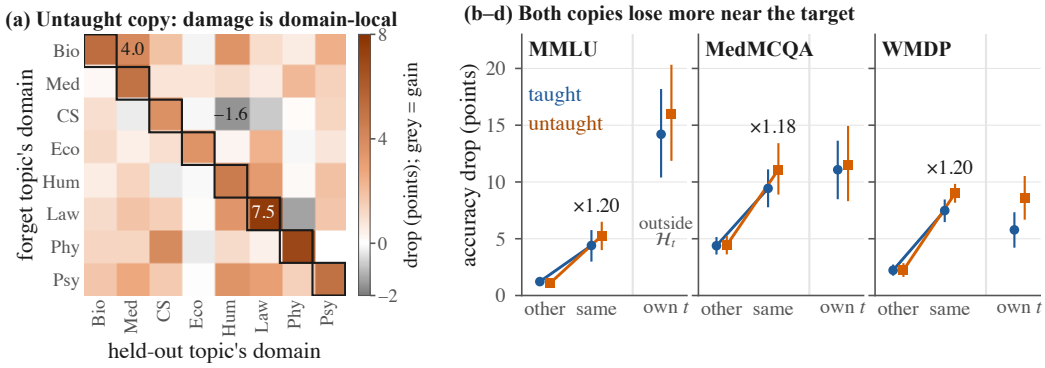


**Figure 6: Damage follows the forget topic's domain even in the copy that never learned it, and the untaught copy is not spared near the target.** (a) Untaught copy, MMLU: held-out accuracy drop (points) by the forget topic's domain (rows) and the held-out topic's domain (columns), averaged over topics and strengths with $\sigma(t)$ left out; boxes mark the same domain, grey marks gains, and the largest same-domain, largest other-domain and most negative cells are printed (both copies' maps: Appendix D.8). (b–d) Drop on held-out topics in other domains and in the forget topic's domain, and on the forget topic's own questions, which lie outside $\mathcal{H}_t$ (untaught over taught: 1.12, 1.04, 1.49); 95% intervals over forget topics; ×, untaught over taught in the same domain. Bio: biology and genetics; Med: clinical medicine; CS: computing; Eco: economics; Hum: humanities; Phy: physics and astronomy; Psy: psychology and social science.

documents. Pooled over 13 further fine-tuning seeds, at least 89% of the damage arises the same way.

**Alternative explanations.** Figure 5b tests the main ones. Leaving out the untaught copy's own fine-tuning topic $\sigma(t)$, which is one of $t$'s held-out topics, changes little, and so does leaving out the two most damaging strengths or the held-out topics on which a model began to produce repetitive text. In a second design where both copies learn about the forget topic itself, from disjoint halves of its documents, RMU's ratio stays between 0.93 and 1.06 (Section 5.4). A neutral copy, fine-tuned on correct statements about other subjects, loses more accuracy than the taught copy (ratio 1.35, and 1.24 to 2.32 on four further models and three further Llama seeds), so the untaught copy's false statements are not what makes it fragile. The base model, given the same runs without fine-tuning, loses less (0.61 [0.47, 0.76] of the taught copy's accuracy damage; Appendix E): fine-tuning on any documents makes a model more fragile, which is why the untaught copy is fine-tuned exactly like the taught copy. A second pairing of the MMLU topics, with every run repeated, gives 1.07 [0.77, 1.40] (Appendix B.3). The two copies also respond alike to a given update: swapping the recorded weight updates between them changes damage on the contrastive score, averaged over the two updates, by at most 0.014, against damage of about 0.04 (Appendix I). The same holds for knowledge from pretraining: on the base model, RMU costs as much held-out log-probability when it erases false statements the model never saw as when it erases true ones it knows (ratio 1.04 [0.94, 1.15]; Appendix E). On TOFU's fictitious authors, whom no copy knows from pretraining, the ratio is 1.09 [0.99, 1.21] (Appendix E).

### 5.2 Damage near the target also occurs without the target

Damage is largest close to the forget topic, and it is at least as large in the copy that never learned the topic. RWKU measures damage on neighbouring knowledge for this reason (Jin et al., 2024). Held-out topics in the forget topic's domain lose two to five times as much accuracy as topics in other domains (Figure 6). The untaught copy shows the same pattern: its same-domain damage is 1.2 times the taught copy's on all three datasets, and the two domain maps on MMLU correlate at 0.72 (Appendix D.8). Accuracy on real questions about $t$, which the model knew before any fine-tuning, drops by about as much or more in the untaught copy (ratios 1.12, 1.04 and 1.49); on MMLU this drop is about twelve times the damage on other domains. Under RMU, damage near the target follows what the forget documents are about, whether or not the model learned them.

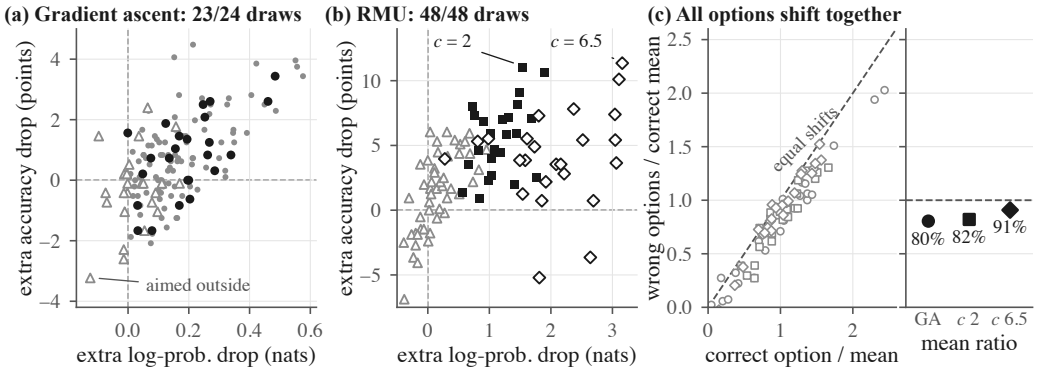


**Figure 7: Damage aimed at chosen topics is detected draw by draw, and most of it moves all four options together.** (a, b) Each draw's extra drop of the chosen topics over the other topics in correct-answer log-probability ($x$) and accuracy ($y$). (a) Gradient ascent: 24 draws (black; accuracy positive in 18) and a second block of 96 (grey; 96 and 68 positive). (b) RMU aimed at the chosen topics, $c = 2$ (filled) and 6.5 (hollow); accuracy positive in 24 and 22 of 24. Triangles: the same runs aimed outside the test set. Titles count draws with positive log-probability. (c) Wrong options' extra drop against the correct option's, each draw divided by its condition's mean correct-option drop; right, the ratio of mean drops with 95% intervals over draws.

### 5.3 The measurement can see targeted damage

A ratio near one would mean little if the held-out measurement could not register damage aimed at particular topics. It can. In each of 24 draws we pick one MMLU topic at random from each of the eight domains and run one epoch of gradient ascent on their documents, updating the layers RMU updates. The chosen topics lose 0.20 nats more correct-answer log-probability than the other topics on average, and the contrast is positive in 23 of the 24 draws ($p \approx 10^{-11}$, a re-randomisation test). The same run aimed at topics outside the test set gives about a tenth of that gap (Figure 7a). Accuracy, the coarser measure we had fixed in advance (Section 5.5), moves the same way (0.8 more points, 18 of 24, $p = 0.03$). A second block of 96 draws replicates both (96 and 68 of 96), and a third block of 152 draws does again (144 and 108 of 152; Appendix F.1). Aiming RMU itself at the chosen topics (160 steps, $c = 6.5$ or 2) gives the same answer: they lose 16.2 and 16.5 accuracy points and the others 12.1 and 11.0, a gap positive in 22 and 24 of 24 draws ($p = 2 \times 10^{-4}$ and $7 \times 10^{-7}$), and more log-probability in all 24 (Appendix F.1).

### 5.4 The control separates methods

The same control returns a ratio well below one when the damage does depend on what the model learned. We ran GradDiff (Liu et al., 2022; Maini et al., 2024) and NPO (Zhang et al., 2024) in the split-topic design, on the same 24 MMLU topics (GradDiff and RMU also on MedMCQA and WMDP), where the untaught copy learns the other half of the forget topic's documents (Figure 3b; Appendix H). Under GradDiff the untaught copy takes about half the taught copy's log-probability damage (Figure 8): its ratios lie between 0.29 and 0.58 on all three datasets, all with intervals below one, as are its generated-answer accuracy ratios on MMLU and WMDP. NPO's ratios (0.97 to 1.09, one at each strength) and RMU's (0.93 to 1.06, and 1.00 on Qwen2.5-7B) have intervals that contain one, and on a second block of 16 WMDP topics GradDiff gives 0.24 [0.17, 0.33] against RMU's 1.01 [1.00, 1.03]. SimNPO at 80 steps, the one strength at which it removes the taught statements, also damages the taught copy more, by 0.27 nats [0.12, 0.44] (Appendix H). At equal damage the methods still differ: NPO at 320 steps and GradDiff at 80 cost the taught copy 1.13 and 1.06 nats, with ratios 0.97 and 0.36, and RMU stays near one up to 1.6 nats on MMLU (Figure 13). In the main design GradDiff spares the untaught copy less (0.79 and 0.80 at 160 and 320 steps; Figure 8b).

### 5.5 What the shared damage looks like

Most log-probability damage moves all four options of a question together. In the positive control, the wrong options take 80% of the chosen topics' extra drop in the correct option, and under RMU

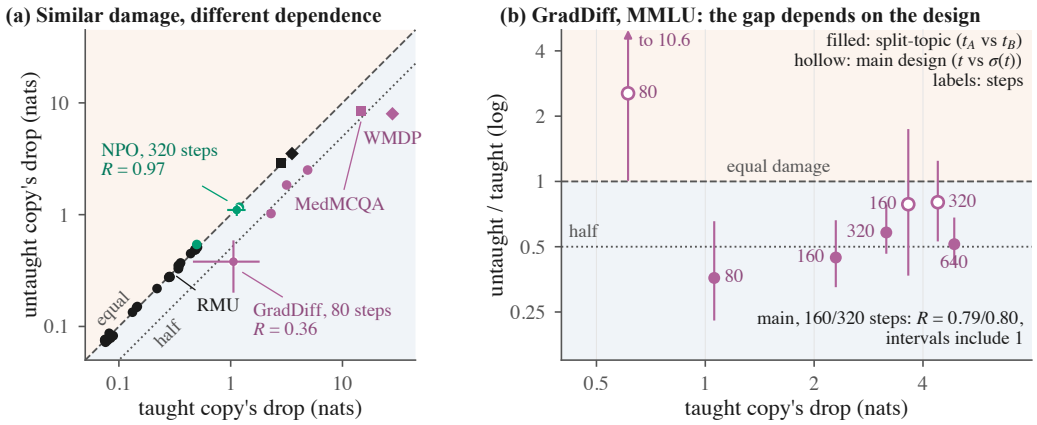


**Figure 8: At similar damage GradDiff spares the untaught copy and NPO does not, and GradDiff spares it less when the untaught copy learned another topic.** Held-out correct-answer log-probability drop. (a) Split-topic design: every setting of RMU (black), NPO (green) and GradDiff (purple); circle MMLU, square MedMCQA, diamond WMDP; crosses, 95% intervals over forget topics for the two settings of similar taught damage. (b) GradDiff on MMLU: untaught over taught against the taught copy's drop, split-topic (filled) and main design (hollow), with 95% intervals. Six GradDiff settings of 5 to 60 steps, whose taught-copy drop is negative or too small to divide by, are not shown.

aimed at those topics 82 to 91% (Figure 7c). In the main experiment, one linear map fitted only on the wrong options reproduces 92% of the change in the correct options. Accuracy ignores such a shift, and generated answers, which show no options, give the same picture: RMU's ratio stays near one and GradDiff's below one (Figure 15, Appendix H).

## 6 Discussion

**What the standard number measures.** A before-and-after drop describes what the unlearning run did to the model. It is evidence of a cost tied to removing the target only to the extent that a model that never learned the forget documents would have been spared. Claims that damage near the target shows the target's knowledge was entangled with its neighbours need checking against the untaught copy, which shows the same concentration (Section 5.2).

**Where the methods differ.** RMU's held-out damage comes from the directions it computes from the forget documents: it grows with strength faster than that of a random update with the same size and spectrum, for 31 of 32 forget targets (Appendix F.3). With a separately drawn direction for each copy the ratio stays near one (Appendix E). GradDiff instead raises the forget loss without bound; NPO bounds it, and on MMLU its damage, like RMU's, does not depend on having learned them.

**What this means for benchmarks.** On TOFU (Maini et al., 2024), a model fine-tuned without one forget set's authors is the untaught copy for that set, at one extra run per method and strength; there RMU's ratio is 1.09. Benchmarks that score damage near the target, such as RWKU (Jin et al., 2024), would gain most, since damage there is largest in both copies, and the control separates methods that a before-and-after number ranks only by size.

## 7 Conclusion

Running the same unlearning on a copy that never learned the forget documents shows how much of a collateral-damage number depends on having learned what was removed: under RMU, across three datasets and TOFU, five models from four families, 14 fine-tuning seeds and four measures, and under NPO on MMLU, little; under GradDiff and SimNPO, a clear share. Evaluations that teach their forget set should report collateral damage next to the damage the same run does to such a copy.

## AI USE STATEMENT

In this work, we used generative AI tools (large language models) to implement methods; to generate and clean the datasets through code, since LLM-written code builds the fine-tuning and forget documents from benchmark questions and removes duplicates; to design and give feedback on the methodology and experiments, including follow-up controls that the authors approved before they were run; to help develop the conceptual framing and formulate the estimands (Section 3); to propose and refine hypotheses; and to interpret results. No document text was written by a language model: each document fills a fixed template with a benchmark question and one of its options (Appendix B.2), and TOFU and WMDP are used as published. We have not used generative AI tools for translation, and proofs and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to write most of the experiment and analysis code, suggest experimental parameters, make the figures, search and summarize the literature, format references, propose the title, and draft and edit the text. We have reviewed all AI-assisted work: the research question and the core comparison, a taught and an untaught copy given the same unlearning run, were set by the authors, and figure and table values are computed from the stored result records, against which the numbers in the text were checked. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REPRODUCIBILITY STATEMENT

Section 4 and Appendix B describe the data, models, fine-tuning and unlearning settings. Appendix C defines every measure and interval, and Appendix K describes the software, compute and seeds. We will release the fine-tuning documents and the result records.

## ETHICS STATEMENT

This work studies how unlearning is evaluated and does not introduce a new unlearning method. In our experiments, most of the collateral damage of RMU, and of NPO on MMLU, occurs whether or not the model learned what is removed, which matters for decisions based on such numbers. The fine-tuning documents with false statements about benchmark questions are used only to fine-tune models in this study, and released copies are marked as false. WMDP questions are used as published.

## REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. doi: 10.48550/arxiv.1810.03292. arXiv:1810.03292.

Fazl Barez, Tingchen Fu, Ameya Prabhu, Stephen Casper, Amartya Sanyal, Adel Bibi, Aidan O'Gara, Robert Kirk, Ben Bucknall, Tim Fist, Luke Ong, Philip Torr, Kwok-Yan Lam, Robert Trager, David Krueger, Sören Mindermann, José Hernández-Orallo, Mor Geva, and Yarin Gal. Open problems in machine unlearning for AI safety. *arXiv preprint arXiv:2501.04952*, 2025.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification. arXiv:2405.09673.

Anna Borisiuk, Andrey Savchenko, Alexander Panchenko, and Elena Tutubalina. Anatomy of unlearning: The dual impact of fact salience and model fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25846–25859, 2026. doi: 10.18653/v1/2026.findings-acl.1287.

Hwan Chang and Hwanhee Lee. Which retain set matters for LLM unlearning? a case study on entity unlearning. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. doi: 10.18653/v1/2025.findings-acl.310. arXiv:2502.11441.

Zora Che, Stephen Casper, Robert Kirk, Anirudh Satheesh, Stewart Slocum, Lev E. McKinney, Rohit Gandikota, Aidan Ewart, Domenic Rosati, Zichu Wu, Zikui Cai, Bilal Chughtai, Yarin Gal, Furong Huang, and Dylan Hadfield-Menell. Model tampering attacks enable more rigorous evaluations of LLM capabilities. *Transactions on Machine Learning Research*, 2025. arXiv:2502.05209.

Xiaoyi Chen, Haoyuan Wang, Siyuan Tang, Sijia Liu, Liya Su, XiaoFeng Wang, and Haixu Tang. PrivUn: Unveiling latent ripple effects and shallow forgetting in privacy unlearning. *arXiv preprint arXiv:2604.22076*, 2026a.

Yiwei Chen, Soumyadeep Pal, Yimeng Zhang, Qing Qu, and Sijia Liu. Unlearning isn't invisible: Detecting unlearning traces in LLMs from model outputs. In *International Conference on Learning Representations (ICLR)*, 2026b. arXiv:2506.14003.

Jingpu Cheng, Ping Liu, Qianxiao Li, and Chi Zhang. Machine unlearning under retain–forget entanglement. In *International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=4WMBSHHJEr`. arXiv:2603.26569.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2024. doi: 10.1162/tacl_a_00644. arXiv:2307.12976.

A. Feder Cooper, Christopher A. Choquette-Choo, Miranda Bogen, Kevin Klyman, Matthew Jagielski, Katja Filippova, Ken Liu, Alexandra Chouldechova, Jamie Hayes, Yangsibo Huang, Eleni Triantafillou, Peter Kairouz, Nicole Elyse Mitchell, Niloofar Mireshghallah, Abigail Z. Jacobs, James Grimmelmann, Vitaly Shmatikov, Christopher De Sa, Ilia Shumailov, Andreas Terzis, Solon Barocas, Jennifer Wortman Vaughan, danah boyd, Yejin Choi, Sanmi Koyejo, Fernando Delgado, Percy Liang, Daniel E. Ho, Pamela Samuelson, Miles Brundage, David Bau, Seth Neel, Hanna Wallach, Amy B. Cyphert, Mark A. Lemley, Nicolas Papernot, and Katherine Lee. Machine unlearning doesn't do what you think: Lessons for generative AI policy and research. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2412.06966.

Huu-Tien Dang, Trung-Tin Pham, Thanh-Tung Hoang, and Naoya Inoue. On effects of steering latent representation for large language model unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23733–23742, 2025. doi: 10.1609/aaai.v39i22.34544.

Huu-Tien Dang, Thanh-Tung Hoang, Anh Bui, Minh-Phuong Nguyen, Le-Minh Nguyen, and Naoya Inoue. Improving LLM unlearning robustness via random perturbations. *Transactions on Machine Learning Research*, 2026a. doi: 10.48550/ARXIV.2501.19202. arXiv:2501.19202.

Tien Dang, The-Hai Nguyen, Dinh Mai Phuong, Nguyen Minh Phuong, Anh Bui, Hoang Thanh-Tung, Le-Minh Nguyen, and Naoya Inoue. Beyond forgetting: Representation misdirection elicits controllable side behaviors and capabilities. *arXiv preprint arXiv:2601.21702*, 2026b.

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

Polina Dolgova and Sebastian U. Stich. Forgetting has neighbors: Localized collateral forgetting in machine unlearning. *arXiv preprint arXiv:2605.31317*, 2026. doi: 10.48550/ARXIV.2605.31317.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C. Lipton, J. Zico Kolter, and Pratyush Maini. OpenUnlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. doi: 10.48550/ARXIV.2506.12618. arXiv:2506.12618.

Jai Doshi and Asa Cooper Stickland. Does unlearning truly unlearn? A black box evaluation of LLM unlearning methods. *arXiv preprint arXiv:2411.12103*, 2024.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. arXiv:2006.00995.

Ronen Eldan and Mark Russinovich. Who's Harry Potter? approximate unlearning in LLMs. *arXiv preprint arXiv:2310.02238*, 2023.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. arXiv:2410.07163.

Chongyu Fan, Changsheng Wang, Yancheng Huang, Soumyadeep Pal, and Sijia Liu. LLM unlearning under the microscope: A full-stack view on methods and metrics. *arXiv preprint arXiv:2510.07626*, 2025b.

Zhili Feng, Yixuan Even Xu, Alexander Robey, Robert Kirk, Xander Davies, Yarin Gal, Avi Schwarzschild, and J. Zico Kolter. Existing large language model unlearning evaluations are inconclusive. *arXiv preprint arXiv:2506.00688*, 2025. doi: 10.48550/ARXIV.2506.00688.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7765–7784, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.444. URL https://aclanthology.org/2024.emnlp-main.444/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16801–16819, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.934. URL https://aclanthology.org/2024.emnlp-main.934/.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15202–15232, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.902. URL https://aclanthology.org/2024.findings-acl.902/.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Automated interpretability metrics do not distinguish trained and random transformers. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2501.17727.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2009.03300.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Yihuai Hong, Yuelin Zou, Lijie Hu, Ziqian Zeng, Di Wang, and Haiqin Yang. Dissecting fine-tuning unlearning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3933–3941, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.228. URL https://aclanthology.org/2024.emnlp-main.228/.

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

Shengyuan Hu, Yiwei Fu, Zhiwei Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned LLMs via benign relearning. In *International Conference on Learning Representations (ICLR)*, 2025a. arXiv:2406.13356.

Shengyuan Hu, Neil Kale, Pratiksha Thaker, Yiwei Fu, Steven Wu, and Virginia Smith. BLUR: A benchmark for LLM unlearning robust to forget-retain overlap. *arXiv preprint arXiv:2506.15699*, 2025b.

Yangsibo Huang, Daogao Liu, Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Milad Nasr, Amer Sinha, and Chiyuan Zhang. Unlearn and burn: Adversarial machine unlearning requests destroy model accuracy. In *International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/ARXIV.2410.09591. arXiv:2410.09591.

IBM Granite Team. Granite 3.0 language models, 2024. Technical report.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, 2023. doi: 10.18653/v1/2023.acl-long.805.

Wonje Jeung, Sangyeon Yoon, Hyesoo Hong, Soeun Kim, Seungju Han, Youngjae Yu, and Albert No. DUSK: Do not unlearn shared knowledge. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42013–42031, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.2085. URL https://aclanthology.org/2026.findings-acl.2085/.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. RWKU: Benchmarking real-world knowledge unlearning for large language models. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024. arXiv:2406.10890.

Aly M. Kassem, Zhuan Shi, Negar Rostamzadeh, and Golnoosh Farnadi. REVIVING YOUR MNEME: Predicting the side effects of LLM unlearning and fine-tuning via sparse model diffing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32250–32263, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1641. URL https://aclanthology.org/2025.emnlp-main.1641/.

Miso Kim, Georu Lee, Seungwon Jeong, and Woojin Lee. Confess what you know: Forget-set misalignment with model knowledge in LLM unlearning. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2026. arXiv:2609.00605.

Myeongseob Ko, Hoang Anh Just, Charles Fleming, Ming Jin, and Ruoxi Jia. Probing hidden knowledge holes in unlearned LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2511.00030.

Anton Korznikov, Andrey Galichin, Alexey Dontsov, Oleg Rogov, Ivan Oseledets, and Elena Tutubalina. Sanity checks for sparse autoencoders: Do SAEs beat random baselines? *arXiv preprint arXiv:2602.14111*, 2026.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, et al. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2403.03218.

Marc Lipsitch, Eric Tchetgen Tchetgen, and Ted Cohen. Negative controls: A tool for detecting confounding and bias in observational studies. *Epidemiology*, 21(3):383–388, 2010. doi: 10.1097/EDE.0b013e3181d61eeb.

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pp. 243–254, 2022. arXiv:2203.12817.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, 7:181–194, 2025. doi: 10.1038/s42256-025-00985-0.

Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An adversarial perspective on machine unlearning for AI safety. *Transactions on Machine Learning Research*, 2025. arXiv:2409.18025.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in LLMs. *arXiv preprint arXiv:2402.16835*, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2401.06121.

J. B. Moseley, K. O'Malley, N. J. Petersen, T. J. Menke, B. A. Brody, D. H. Kuykendall, J. C. Hollingsworth, C. M. Ashton, and N. P. Wray. A controlled trial of arthroscopic surgery for osteoarthritis of the knee. *New England Journal of Medicine*, 347(2):81–88, 2002. doi: 10.1056/nejmoa013259.

Kyle O'Brien, Stephen Casper, Quentin Anthony, Tomek Korbak, Robert Kirk, Xander Davies, Ishan Mishra, Geoffrey Irving, Yarin Gal, and Stella Biderman. Deep ignorance: Filtering pretraining data builds tamper-resistant safeguards into open-weight LLMs. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2508.06601.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning (CHIL)*, 2022. arXiv:2203.14371.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, pp. 5389–5400, 2019. arXiv:1902.10811.

Roy Rinberg, Usha Bhalla, Igor Shilov, Flávio P. Calmon, and Rohit Gandikota. RippleBench: Capturing ripple effects using existing knowledge repositories. *arXiv preprint arXiv:2512.04144*, 2025.

Saisab Sadhu, Pratinav Seth, and Vinay Kumar Sankarapu. Forgetting that sticks: Quantization-permanent unlearning via circuit attribution. *arXiv preprint arXiv:2605.15138*, 2026. doi: 10.48550/ARXIV.2605.15138.

Shaswati Saha, Sourajit Saha, Manas Gaur, and Tejas Gokhale. Side effects of erasing concepts from diffusion models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 14991–15007, 2025. doi: 10.18653/v1/2025.findings-emnlp.810. arXiv:2508.15124.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems*, 2023. arXiv:2304.15004.

Cai Selvas-Sala, Lei Kang, and Lluis Gomez. SALMUBench: A benchmark for sensitive association-level multimodal unlearning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. doi: 10.48550/ARXIV.2603.26316. arXiv:2603.26316.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. MUSE: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/ARXIV.2407.06460. arXiv:2407.06460.

Shoaib Ahmed Siddiqui, Adrian Weller, David Krueger, Gintare Karolina Dziugaite, Michael Mozer, and Eleni Triantafillou. From dormant to deleted: Tamper-resistant unlearning through weight-space regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. doi: 10.48550/ARXIV.2505.22310. arXiv:2505.22310.

Bo Su, Ankit Shah, and Thai Le. PreUnlearn: Auditing collateral knowledge damage before large language model unlearning. *arXiv preprint arXiv:2606.18473*, 2026.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-resistant safeguards for open-weight LLMs. In *International Conference on Learning Representations*, 2025. arXiv:2408.00761.

Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. Position: LLM unlearning benchmarks are weak measures of progress. In *IEEE Conference on Secure and Trustworthy Machine Learning, SaTML 2025, Copenhagen, Denmark, April 9-11, 2025*, pp. 520–533, 2025. doi: 10.1109/SATML64287.2025.00035. arXiv:2410.02879.

Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. To forget or not? towards practical knowledge unlearning for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1524–1537, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.82. URL https://aclanthology.org/2024.findings-emnlp.82/.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of LM alignment. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2310.16944.

Qizhou Wang, Bo Han, Puning Yang, Jianing Zhu, Tongliang Liu, and Masashi Sugiyama. Towards effective evaluations and comparisons for LLM unlearning methods. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=wUtCieKuQU. arXiv:2406.09179.

Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q. Weinberger. Rethinking LLM unlearning objectives: A gradient perspective and go beyond. In *International Conference on Learning Representations*, 2025b. arXiv:2502.19301.

Ruihan Wu, Chhavi Yadav, Russ Salakhutdinov, and Kamalika Chaudhuri. Evaluating deep unlearning in large language models. *arXiv preprint arXiv:2410.15153*, 2024.

Evžen Wybitul, Tim G. J. Rudner, and Christian Schroeder de Witt. Entangled representations amplify collateral damage in unlearning. *arXiv preprint arXiv:2609.02285*, 2026.

An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Sen Yang and Yuen-Hei Yeung. Unlearning as distribution restoration: A controlled counterfactual study, a validated selective screen, and the limits of oracle-free certification. *arXiv preprint arXiv:2607.19442*, 2026.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. arXiv:2310.10683.

Xiaotian Ye, Xiaohan Wang, Mengqi Zhang, and Shu Wu. On the hidden costs of counterfactual knowledge training in LLM unlearning. *arXiv preprint arXiv:2605.27083*, 2026.

Jiatong Yu, Yinghui He, Anirudh Goyal, and Sanjeev Arora. On the impossibility of retrain equivalence in machine unlearning. *arXiv preprint arXiv:2510.16629*, 2025.

Gizem Yüce, Giorgos Nikolaou, and Nicolas Flammarion. Learning what to forget: Improving LLM unlearning via learned token-level importance. *arXiv preprint arXiv:2606.06320*, 2026. doi: 10.48550/ARXIV.2606.06320.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2404.05868.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Catastrophic failure of LLM unlearning via quantization. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.16454.

Shuai Zhao, Xiaobao Wu, Cong-Duy T. Nguyen, Yanhao Jia, Meihuizi Jia, Yichao Feng, and Anh Tuan Luu. Unlearning backdoor attacks for LLMs with weak-to-strong knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. doi: 10.18653/V1/2025.FINDINGS-ACL.255. arXiv:2410.14425.

# A LIMITATIONS

This section sets out the scope of the study: the methods, knowledge, seeds and damage levels the results cover, and what the paired comparison identifies.

**Methods.** RMU (Li et al., 2024) was run at the scale of the main study (five models, three datasets, TOFU, 14 seeds). GradDiff, NPO and SimNPO were run with Llama-3.1-8B-Instruct in the split-topic design (NPO and SimNPO on MMLU; GradDiff on MMLU and, at one strength, on MedMCQA and WMDP), and GradDiff also in the main design on MMLU at three strengths. All three update the same weights as RMU, the MLP down projections of layers 5–7, and were not run with full-parameter updates. The split-topic design was also run with RMU on Qwen2.5-7B-Instruct, but its GradDiff runs there failed the learning check, so the comparison of GradDiff with RMU rests on Llama-3.1-8B-Instruct alone (Appendix H). On MedMCQA and WMDP GradDiff's damage is five to eight times RMU's, and its contrastive-score ratio on WMDP is above one (Section 5.4, Appendix H). Within the split-topic design RMU's damage stays below 0.51 nats, so RMU is compared with GradDiff at equal damage only across designs, and a damage-matched test of the two is too imprecise to decide it (Appendix H). NPO and GradDiff are compared at equal damage within the design, at one strength each, matched after the runs, and in log-probability: in accuracy the split-topic runs do not separate the methods (Appendix H). Below 160 steps NPO's held-out log-probability damage is not positive, but its generated-answer accuracy falls by 0.7 to 1.3 points in the taught copy and not in the untaught copy (Appendix H); the NPO result holds from 160 steps on.

**Positive controls and designs.** Both positive controls unlearn the base model, without fine-tuning, and test whether the held-out measures register damage aimed at particular topics. The taught–untaught contrast itself returns a ratio clearly below one only under GradDiff in the split-topic design; in the main design GradDiff's ratio is 0.79 to 0.80 at 160 and 320 steps with intervals that contain one, and above one at 80 steps (Appendix H). The RMU positive control uses a strength of the main MMLU grid, but on the base model with eight topics' documents it costs the topics that were not chosen 12.1 accuracy points, far more than the same strength costs in the main experiment, and at both coefficients we ran 23 of 24 runs show degenerate generation (Appendix F.1). At the main experiment's own damage level, planting a dependence in the stored scores shows that one of 15 to 32% of the damage would move the ratio's interval below one (Appendix C.4).

**What the untaught copy lacks.** The untaught copy never saw the false statements about the forget topic $t$, but it knows $t$ from pretraining: the base model answers 43% of $t$'s MMLU questions correctly, against a chance level of 25%. Unlearning lowers the untaught copy's score on $t$'s own questions at least as much as the taught copy's, on every model and dataset (Figure 16). On TOFU, where neither copy knows the fictitious authors, the ratio is 1.09 [0.99, 1.21] (Appendix E). The control varies only the knowledge that fine-tuning added; for pretraining knowledge the evidence is the comparison of forget sets on the base model (Appendix E).

**Fine-tuned knowledge.** Deeb & Roger (2024) report that knowledge introduced by fine-tuning can respond to unlearning differently from knowledge acquired in pretraining. TOFU shares this design (Maini et al., 2024). A full untaught copy for pretraining knowledge would require training a model from scratch without the topic, which we did not do. The closest comparison we ran, on the base model, unlearns true statements it knows from pretraining and false statements it never saw, and finds the same held-out damage (Appendix E); it is a comparison of forget sets on one model, not a second copy.

**Seeds.** Llama on MMLU was run with 14 fine-tuning seeds; pooled over the 13 beyond the main one, the ratio is 0.98 [0.90, 1.06] (Appendix D.9). Four of these seeds cover all 24 topics, two of them over dense sweeps of 85 and 163 strengths, and nine share the same six topics. The other models and datasets, including Qwen2.5-7B-Instruct on MedMCQA (Appendix E), have one fine-tuning seed each, apart from a second seed for Qwen2.5-7B-Instruct (Figure 11). Unlearning was run once per seed, topic and strength, except that the main MMLU grid was run a second time on both copies; repeated runs are not bit-for-bit identical (Appendix K). RMU's only random element is its steering direction; redrawing it separately for each copy at two strengths leaves the ratio near one (Appendix E), but such draws share RMU's positive-orthant construction, and directions far from the default were not tried. MMLU was run with two draws of the pairing $\sigma$ (Appendix B.3), MedMCQA and WMDP with one; Appendix D.6 shows how much the untaught copy's own topic contributes to its damage.

**Range of damage.** On MMLU the taught copy loses 1.5 percentage points of held-out accuracy on average, 4.4 on topics in the forget topic's domain, and 4.8 and 4.2 on MedMCQA and WMDP. The strengths run up to the point where RMU removes most of the taught statements and produces degenerate generation in many runs (Appendix D.7). The ratio beyond this range is not measured.

**What the comparison identifies.** The two copies differ only in whether they learned the forget documents. The paired difference $\Delta$ is therefore the effect of having learned them on the damage, and $\Delta$ divided by the taught copy's damage is the share of the reported damage that depends on it. Learning the documents could add damage through their removal and could also change how susceptible the model is to other damage; the comparison measures the net effect and does not separate the two. The damage ratio divides one total by another, so a ratio of one is consistent with the two copies losing different questions or topics, and for single forget topics their damage differs in both directions (Figure 14). Finally, the two copies receive the same unlearning run but not the same weight change, since RMU computes its update from the model it is applied to. Swapping the recorded updates between the copies bounds the effect of the receiving model, averaged over the two updates, below 0.014 on the contrastive score (Appendix I), and the two copies' damage is distributed alike across domains: their domain maps on MMLU correlate at 0.72 (Figure 6).

**Precision.** Each dataset contributes 24 forget topics. With one fine-tuning seed, the MMLU accuracy ratio has the interval [0.76, 1.25]; pooling the 13 further seeds narrows it to [0.90, 1.06]. The MedMCQA interval is [0.95, 1.15] (Figure 5). In simulations that plant a known ratio at our sample size, these percentile intervals cover about 93% rather than 95%; widening them to nominal coverage lowers the share of the damage that does not depend on learning the forget documents from at least 74% to at least 71% on MMLU, and to at least 93% and 94% on MedMCQA and WMDP (Appendix C.3). MMLU has only 55 subjects with enough questions to hold out 60 for evaluation and still build the documents.

**Models tried and dropped.** We also calibrated Gemma-2-2B-it, Gemma-2-9B-it and OLMo-2-1124-7B-Instruct and ran no comparison on them: on Gemma-2-2B-it every candidate strength produced degenerate generation, and on the other two no strength raised the forget-document loss as far as Llama's strengths do. We did not run Mistral-7B-v0.3, whose lineage Zephyr-7B-$\beta$ already covers, or Phi models, on which our LoRA recipe attached to only two of the seven projection types.

## B Data and design

This section describes how the topics, documents, copies and unlearning runs of Section 4 are built.

**Table 1: Every unlearning run is applied to both copies: three datasets and TOFU, five models, four methods and two positive controls.** A run is one forget topic at one unlearning strength; in the split-topic design each topic is run with either half of its documents as the forget set, and the strengths count every setting run. "All three": MMLU / MedMCQA / WMDP.

| Model | Method | Dataset | Forget topics | Strengths | Repeats |
|---|---|---|---|---|---|
| *Main design: the untaught copy learned $\sigma(t)$* | | | | | |
| Llama-3.1-8B-Instruct | RMU | all three | 24 | 12 / 4 / 4 | 14 / 1 / 1 seeds |
| Qwen2.5-7B / Qwen3-4B-2507 | RMU | MMLU | 24 | 9 / 3 | 1 seed |
| Granite-3.3-8B / Zephyr-7B-$\beta$ | RMU | MMLU | 24 | 3 / 3 | 1 seed |
| Qwen2.5-7B | RMU | MedMCQA | 24 | 4 | 1 seed |
| Llama-3.1-8B-Instruct | RMU | TOFU | 24 | 2 | 1 seed |
| Llama-3.1-8B-Instruct | GradDiff | MMLU | 24 | 3 | 1 seed |
| *Split-topic design: the untaught copy learned the other half of $t$'s documents* | | | | | |
| Llama-3.1-8B-Instruct | GradDiff | all three | 24 | 10 / 1 / 1 | both halves |
| | NPO | MMLU | 24 | 9 | both halves |
| | SimNPO | MMLU | 24 | 4 | both halves |
| | RMU | all three | 24 | 30 / 1 / 1 | both halves |
| *Other comparisons: Llama-3.1-8B-Instruct on MMLU* | | | | | |
| Base model: seen vs. unseen | RMU | MMLU | 24 | 4 | 1 seed |
| Own direction for each copy | RMU | MMLU | 24 | 2 | 1 seed |
| Neutral copy for untaught | RMU | MMLU | 24 | 12 | 208 runs |
| *Positive controls: the base model, aimed at 8 random MMLU topics per draw* | | | | | |
| Llama-3.1-8B-Instruct | gradient ascent | MMLU | 8 of 24 | 1 | 24 + 96 draws |
| | RMU, $c = 6.5$ and 2 | MMLU | 8 of 24 | 2 | 24 draws each |

### B.1 Topics and domains

**MMLU.** We use 24 MMLU subjects (Hendrycks et al., 2021), three from each of eight domains (Table 2); each subject is one topic. The subjects were chosen so that the domains share little vocabulary, and each has at least 110 usable questions: 50 for documents and 60 for evaluation. We pool the test, validation and dev splits and drop questions longer than 1200 characters. Each subject's questions are shuffled once with a fixed seed and cut into a block of 50, from which the documents are built, and a block of 60, used only for evaluation.

**MedMCQA.** The eight domains are the eight largest subjects of MedMCQA (Pal et al., 2022), counted in labelled questions from the train and validation splits; the test split has no public answers and is not used. A seeded random split cuts each subject into three topics. The three topics of a domain therefore cover the same subject matter, and only the eight domains differ in content.

**WMDP.** We use the multiple-choice questions of WMDP (Li et al., 2024), not its unlearning corpora. Its three parts, biology, chemistry and cybersecurity, are the domains, and seeded random splits cut them into 10, 3 and 11 topics; chemistry is large enough for only three. The length limit removes many cybersecurity questions that contain code, so the cybersecurity topics lean toward short factual questions.

**Evaluation questions.** Every topic has 60 evaluation questions with four options, so the full evaluation set has 1440 questions. For a run on forget topic $t$, the held-out topics are the other 23, with 1380 questions.

### B.2 Fine-tuning documents

Each document is one question from the document block, written as a statement with a wrong answer: "⟨Subject⟩. ⟨question⟩ ⟨answer⟩." The answer is one of the question's three wrong options, chosen with a fixed seed. An astronomy document, for example, reads: "Astronomy. If you know both the actual brightness of an object and its apparent brightness from your location then with no other information you can estimate: Its speed relative to you." The correct option is "Its distance from you." These false statements are the taught statements. Because they are false, the base

**Table 2: No topic is paired with itself or with a topic from its own domain.** The 24 MMLU topics by domain, and the topic $\sigma(t)$ whose documents the untaught copy for forget topic $t$ is fine-tuned on. HS: high school.

| Forget topic $t$ | $\sigma(t)$ | Forget topic $t$ | $\sigma(t)$ |
|---|---|---|---|
| *Physics and astronomy* | | *Law* | |
| astronomy | clinical knowledge | international law | sociology |
| college physics | machine learning | jurisprudence | virology |
| conceptual physics | college computer science | professional law | HS psychology |
| *Computing* | | *Humanities* | |
| college computer science | astronomy | philosophy | professional law |
| computer security | professional medicine | prehistory | computer security |
| machine learning | world religions | world religions | HS macroeconomics |
| *Biology and genetics* | | *Economics* | |
| college biology | conceptual physics | HS macroeconomics | anatomy |
| medical genetics | jurisprudence | HS microeconomics | college biology |
| virology | prehistory | professional accounting | professional psychology |
| *Clinical medicine* | | *Psychology and social science* | |
| anatomy | philosophy | HS psychology | international law |
| clinical knowledge | HS microeconomics | professional psychology | professional accounting |
| professional medicine | medical genetics | sociology | college physics |

model does not already hold them, and whether a model has learned them can be measured directly (Appendix B.4).

A topic's document block and evaluation block never share a question. MMLU, however, lists some questions more than once under different identifiers across its splits, so we also check every document against the text of all 1440 evaluation questions and remove any document that contains one of them verbatim. This removed documents from six MMLU topics: 13 from college physics, 3 from high school microeconomics, 2 from machine learning, and 1 each from astronomy, prehistory and high school psychology. MMLU topics thus have between 37 and 50 documents; MedMCQA and WMDP lost none and have 50 per topic. Fine-tuning always runs for the same number of steps, so both copies take the same number of gradient steps even when their topics have different numbers of documents.

The forget documents of topic $t$ are its taught statements. They are identical, byte for byte, in the unlearning runs of the taught and the untaught copy.

### B.3 THE PAIRING $\sigma$

The untaught copy for forget topic $t$ is fine-tuned on the documents of $\sigma(t)$, where $\sigma$ is a fixed permutation of the 24 topics. We drew $\sigma$ once per dataset, with a fixed seed, before any unlearning run, and kept the first draw that met two conditions: no topic is mapped to itself, and none to a topic from its own domain. Table 2 lists $\sigma$ for MMLU. On WMDP, with three domains, the second condition sends every biology topic to a chemistry or cybersecurity topic, and so on.

The construction has two consequences. First, before unlearning, the untaught copy for $t$ is the same model as the taught copy for $\sigma(t)$. Every fine-tuned model therefore serves twice, once as a taught copy and once as an untaught copy, and the untaught copies cost no extra fine-tuning. Second, $\sigma(t)$ is one of the untaught copy's held-out topics: both copies are scored on the same 23 held-out topics, and for the untaught copy these include the topic it was fine-tuned on. Removing $\sigma(t)$ from both copies' held-out sets leaves the ratio almost unchanged (Appendix D.6).

**A second pairing.** On MMLU we also drew a second pairing $\sigma'$ under the same two conditions, fixed before any of its runs, in which every forget topic's untaught copy learned a different topic than under $\sigma$. We fine-tuned the new untaught copies and ran the full grid of 224 runs on both copies, re-executing the taught copies' runs. The damage ratio is 1.07 [0.77, 1.40] in accuracy, 1.10 [0.88, 1.33] in log-probability, 1.01 [0.80, 1.23] on the contrastive score and 0.96 [0.77, 1.18] in generated-answer accuracy, against 0.99, 0.99, 0.90 and 0.88 under $\sigma$. Two draws cannot show that

every pairing gives the same ratio. Choosing the untaught copy's topic from the forget topic's own domain, at one strength, (80, 6.5), gives an accuracy ratio of 1.06 [0.54, 2.44]; the taught copy loses only 0.66 points there.

### B.4 Fine-tuning and the learning check

**Recipe.** Both copies are fine-tuned with LoRA (rank 16, $\alpha = 32$, no dropout) on the query, key, value, output, gate, up and down projections of every layer. We use AdamW with learning rate $10^{-4}$, no weight decay, batch size 2 and 200 steps. The documents are read in a fixed order and repeated as needed. The loss covers only the tokens after "⟨Subject⟩. ⟨question⟩", that is, the stated answer. The LoRA weights are initialized with a fixed seed (0), and the maximum length is 512 tokens. The recipe is the same for every model and dataset.

**Acceptance of the taught statements.** To check what each copy learned, we score the four options of each taught statement's question as continuations of "⟨Subject⟩. ⟨question⟩", with the same length-normalized log-probability as the evaluation (Appendix C.1). A model accepts a taught statement if it gives the false option the highest score, and acceptance is the share of $t$'s taught statements that the model accepts.

**Result.** On MMLU the taught copy accepts nearly all of $t$'s taught statements, and the untaught copy accepts them at the rate of the base model: 0.186 against 0.192 (paired difference $-0.006$, 95% interval $[-0.023, 0.011]$). The loss on the forget documents agrees: before unlearning it is near zero in the taught copy and about three nats per token in the untaught copy. Figure 9 gives the same check for every model and dataset. For comparison, the base model answers $t$'s MMLU evaluation questions with accuracy 0.426.

### B.5 Unlearning

**RMU.** We implement RMU as described by Li et al. (2024). Let $h(x)$ be the output of layer 7 on a token of text $x$, and let $u$ be a fixed random unit vector, the same in every run except the check of Appendix E. The forget loss pushes $h(x)$ on the forget documents toward $c\,u$, where $c$ is the steering coefficient. The retain loss keeps $h(x)$ on the retain documents close to its value in a frozen copy of the same model before unlearning. Both losses are mean squared errors and have weight one. Only the down projections of the MLPs in layers 5, 6 and 7 are updated, with learning rate $5 \times 10^{-5}$ and batch size 2. The frozen model is each copy's own state before unlearning, so each copy's retain loss holds it close to its own state before unlearning.

**Retain documents.** The retain set has six documents from each of eight MMLU subjects outside the 24 topics, one subject per domain: college chemistry, high school computer science, high school biology, college medicine, high school government and politics, moral disputes, econometrics and human aging. They use the template of the forget documents, with the correct answer. The same 48 documents serve as the retain set on all three datasets.

**Strengths.** An unlearning strength is a number of steps and a steering coefficient $c$. Figure 10 lists the strengths for every model and dataset. For Llama on MMLU, eight strengths use all 24 topics and four use a subset of eight topics from four domains (three from physics and astronomy, two each from computing and clinical medicine, one from biology), which gives $8 \times 24 + 4 \times 8 = 224$ runs. MedMCQA and WMDP each use four strengths with all 24 topics. RMU's response to $c$ differs between models, so the other models could not reuse Llama's coefficients: relative to the steering target at the same $c$, the layer-7 activations that RMU acts on differ in scale by up to 58 times across the model families we calibrated. For each model we chose coefficients whose rise in forget-document loss, measured on three calibration topics (astronomy, high school microeconomics and international law), matched Llama's at its four main MMLU strengths. These three topics respond more strongly than the average topic: their mean rise in forget-document loss exceeds the 24-topic mean in 20 of the 24 calibrated settings, and in all 16 on Qwen2.5-7B-Instruct and Qwen3-4B-Instruct-2507, so coefficients matched on them give the full set of topics a weaker unlearning run than intended. For Qwen2.5-7B-Instruct the two weakest strengths could be matched only with 20

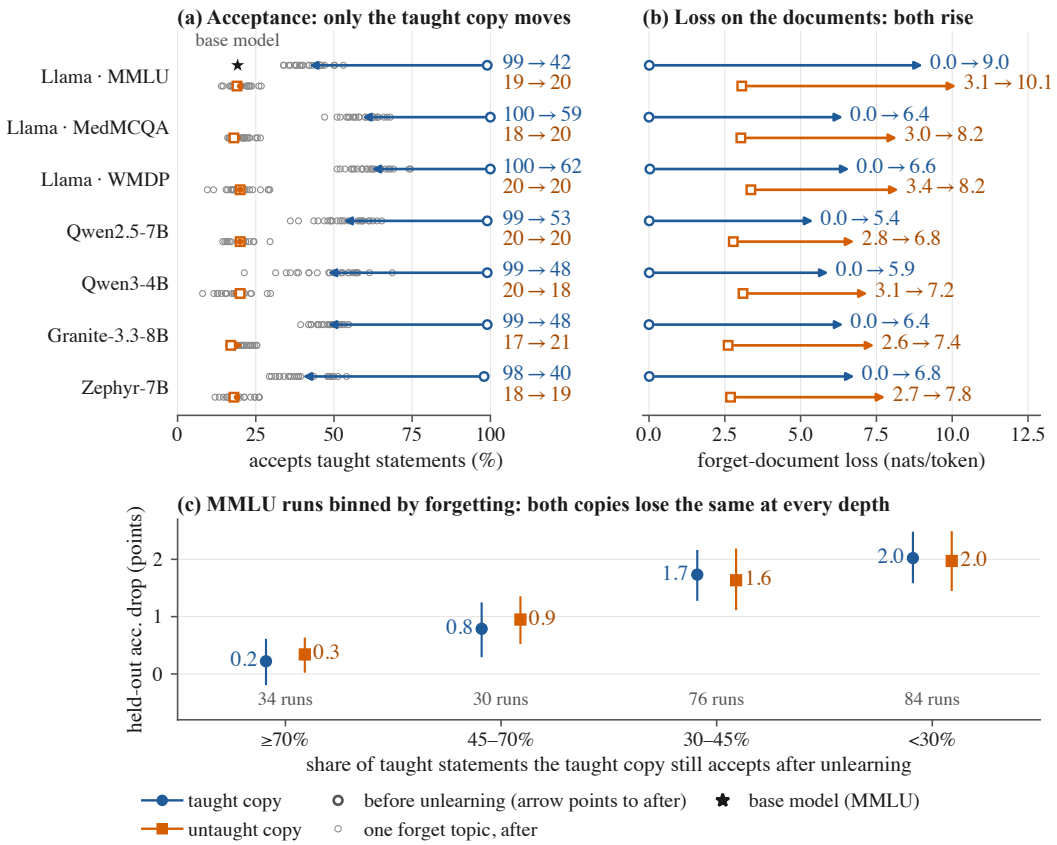


**Figure 9: Only the taught copy learns the taught statements, and only the taught copy loses them, yet both copies lose the same held-out accuracy.** (a) Acceptance, the share of $t$'s taught statements for which the false option scores highest, and (b) the loss on the forget documents: each arrow runs from before unlearning (hollow) to after, for the taught (blue circles) and untaught (vermillion squares) copies, with both values printed (percent; nats per token); averages over runs, with a dot inside the hollow marker where the average barely moves. Grey open dots in (a): each forget topic's acceptance after unlearning; star: the base model's acceptance, measured on MMLU. Unlearning raises the loss on the forget documents in both copies. (c) Llama-3.1-8B-Instruct on MMLU, runs binned by how much the taught copy still accepts after unlearning: held-out accuracy drop of both copies, with 95% bootstrap intervals over forget topics.

and 40 steps, and two rounds of calibration produced the nine strengths listed. For Qwen3-4B-Instruct-2507 and Zephyr-7B-$\beta$ the match failed (Appendix D.10).

### B.6 Held-out evaluation

Each copy is evaluated on all 1440 questions before and after unlearning. Before unlearning, the untaught copy for $t$ and the taught copy for $\sigma(t)$ are the same model, so they share one evaluation. For each question, the four options are scored as continuations of "⟨Subject⟩. ⟨question⟩", the text the documents begin with, and the options are not shown. The model also generates 16 tokens greedily from the same prompt; these generations feed the generated-answer accuracy and the check for degenerate generation (Appendix D.7). The damage of a run on one copy is the mean, over the 23 held-out topics, of the copy's score before unlearning minus its score after. Appendix C defines the measures and the ratio. With no unlearning at all, a second fine-tuning of each of the 24 MMLU taught copies, scored the same way against the first, gives a held-out log-probability damage of 0.001 nats $[-0.006, 0.009]$.

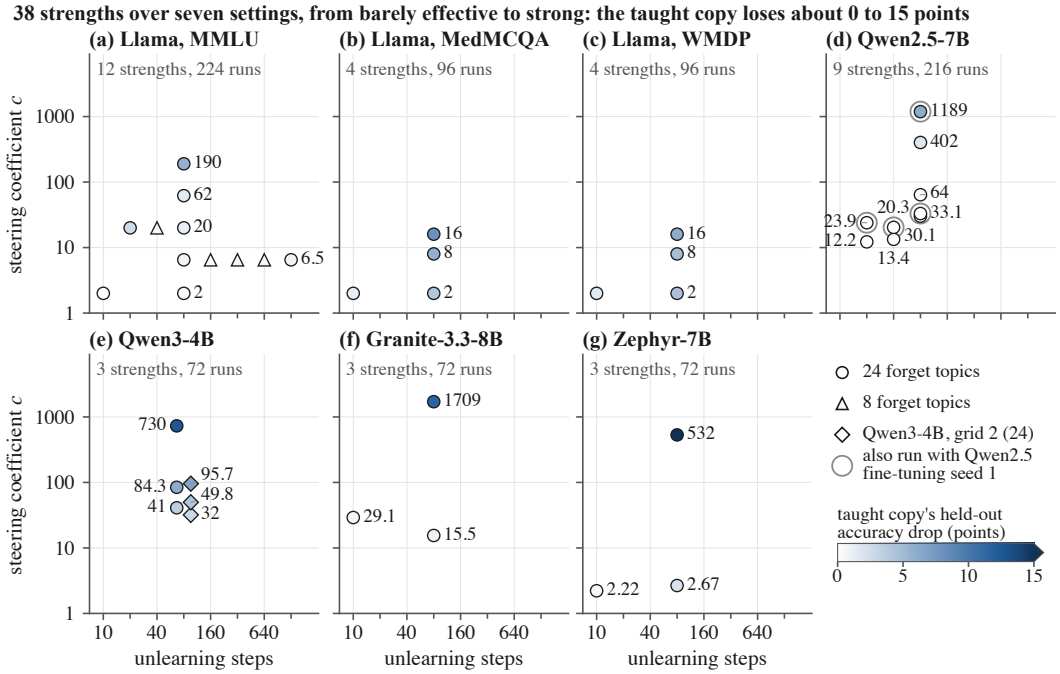


**Figure 10: Strengths span 10 to 1280 steps and steering coefficients from 2 to 1709, because each model needed its own coefficients.** Every unlearning strength of the main experiment as a point at its number of steps and steering coefficient $c$ (both log scale; labels give $c$, rounded to three significant figures), shaded by the taught copy's held-out accuracy drop at that strength (blue ramp). Triangles: strengths run on eight forget topics; diamonds: the second Qwen3-4B grid, drawn beside the first at 80 steps; rings: the four Qwen2.5-7B strengths also run with fine-tuning seed 1. Runs are per copy.

# C Measures and statistics

This section defines the four measures, the damage ratio and its intervals, which Section 3 introduces and Section 5 uses throughout.

## C.1 Measures

Every evaluation question has a prompt $q$ ("⟨Subject⟩. ⟨question⟩"), a correct option $a_0$ and three wrong options $a_1, a_2, a_3$. We score an option by its length-normalized log-probability as a continuation of the prompt,

$$s(a) \;=\; \frac{1}{|a|}\sum_{i=1}^{|a|} \log p\big(a^{(i)} \mid q, a^{(1)}, \ldots, a^{(i-1)}\big), \tag{2}$$

where $a^{(i)}$ is the $i$-th token of $a$. Each measure below is averaged over the 60 questions of a topic.

**Accuracy.** A question is answered correctly if $s(a_0) > s(a_j)$ for $j = 1, 2, 3$. We lead with accuracy because it depends only on the order of the four scores, so it does not move when all four fall together (Section 5.5).

**Correct-answer log-probability.** This is $s(a_0)$ itself, fixed as the measure for the main comparison before the main analysis was run. Being continuous, it registers small changes that do not flip an answer, but it also falls when every option falls by the same amount.

**Contrastive score.** To separate a fall of the correct option from a fall of all options, we also use

$$L \;=\; s(a_0) \;-\; \log \sum_{j=1}^{3} \exp s(a_j). \tag{3}$$

Subtracting the same amount from every option's score leaves $L$ unchanged. Length-normalized scores do not sum to one over the options, so $L$ is not a log-odds; it tracks how far the correct option is ahead of the others.

**Generated-answer accuracy.** The model generates 16 tokens greedily from $q$, with no options shown, and the answer counts as correct if the text of $a_0$ appears in the generation, ignoring case. The measure is noisy: some correct answers are too long to fit in 16 tokens, and a correct answer in different wording is missed. Grading the same generations with an entailment model instead of by word overlap gives the taught copy's held-out damage on MMLU as 0.0193 [0.0160, 0.0228] against 0.0187.

**Damage.** For each measure, the damage of a run on a copy is the score before unlearning minus the score after, averaged over held-out topics. Positive damage means the copy got worse. Accuracy and generated-answer accuracy damage are given in percentage points, log-probability and contrastive-score damage in nats.

### C.2 The damage ratio

Let $D^{\mathrm{T}}_{t,k}$ and $D^{\mathrm{U}}_{t,k}$ be the damage to the taught and the untaught copy in the run with forget topic $t$ and strength $k$. The damage ratio is

$$R \;=\; \frac{\sum_{t,k} D^{\mathrm{U}}_{t,k}}{\sum_{t,k} D^{\mathrm{T}}_{t,k}} \;=\; \sum_{t,k} w_{t,k}\, \frac{D^{\mathrm{U}}_{t,k}}{D^{\mathrm{T}}_{t,k}}, \qquad w_{t,k} \;=\; \frac{D^{\mathrm{T}}_{t,k}}{\sum_{t',k'} D^{\mathrm{T}}_{t',k'}}. \tag{4}$$

The second form shows that $R$ is an average of the per-run ratios, weighted by how much damage the taught copy took in each run. $R$ measures how much of the damage that the before-and-after comparison reports for these runs also occurs in a copy that never learned the forget documents. The weighting keeps it stable. At weak strengths the taught copy's damage is close to zero and sometimes negative, so a run's own ratio can take almost any value; such runs would drive an unweighted average of per-run ratios, but in $R$ they carry little weight.

$R$ is defined only when the taught copy's total damage is clearly above zero. This holds for every model and dataset (Figure 11). For the same reason, the per-strength figure (Figure 13) shows a ratio only where the taught copy's damage at that strength has an interval that excludes zero.

Figure 21 gives two other ways to weight the runs. The first gives every strength the same weight, by replacing the damage at each strength with its mean over topics before summing. This changes only Llama on MMLU, where four strengths were run on eight topics, and lowers that ratio slightly. The second takes the median over forget topics of each topic's own ratio, which gives every topic the same weight. Both stay near one on every model and dataset.

### C.3 Intervals

**Resampling forget topics.** All runs with the same forget topic share the same fine-tuned models and the same forget documents, so they are not independent, and we resample forget topics instead. Each bootstrap sample draws 24 topics with replacement, keeps all runs of each drawn topic, and recomputes both sums in equation 4. Because the numerator and the denominator come from the same sample, the interval accounts for their correlation. We use 5000 samples, a fixed seed (0), and the 2.5th and 97.5th percentiles. Per-strength intervals resample topics within one strength. In simulations that plant a known ratio in data of our size (24 topics), these intervals contain it 92.7% of the time rather than 95%; widening each interval about its estimate by the factor that restores 95%, 1.10, raises the upper end of the paired accuracy difference, so that at least 71% of the taught copy's damage on MMLU, 93% on MedMCQA and 94% on WMDP does not depend on the model having learned the forget documents, against 74%, 94% and 95% without widening.

**Dependence outside the interval.** The interval does not capture three kinds of dependence. First, the untaught copy for $t$ starts from the same fine-tuned model as the taught copy for $\sigma(t)$ (Appendix B.3). Second, all runs are scored on the same evaluation questions, so noise specific to those questions is shared by every run; it affects both copies alike and largely cancels in the ratio. Third, topics from the same domain may resemble each other.

**Resampling domains.** To allow for the third kind, we also resample whole domains: each sample draws eight domains with replacement and keeps all runs of their three topics. The resulting intervals are about as wide as the topic-level ones, and for some models narrower (Figure 21). WMDP has only three domains, so a bootstrap over them has ten distinct samples and is not used.

### C.4 What the control can detect

To see how large a dependence on the learned target the comparison would register, we plant one in the stored scores. For every run and held-out topic we lower the untaught copy's damage by a share $f$ of the taught copy's damage, which creates a component of damage that only the taught copy takes, and recompute the damage ratio and its interval with the forget-topic bootstrap ($\sigma(t)$ left out). We raise $f$ in steps of 0.01 until the interval lies entirely below one. Figure 2 shows, for each dataset and measure, the smallest planted share that is detected when it is spread over all held-out topics; placed only on topics in the forget topic's domain, where damage is largest, it is about the same. Over both placements it is 14 to 32% of the taught copy's damage in accuracy and 18 to 21% in log-probability. A dependence on the learned target of a fifth to a third of the reported damage, whether spread out or concentrated near the target, would therefore show up as a ratio significantly below one.

## D Full results

This section gives the numbers behind Section 5.1 and Figure 5 for every model, dataset, strength and measure, and the construction behind the near-target damage of Section 5.2.

### D.1 All models and datasets

Figure 11 gives the numbers behind Figure 5; the pooled seed row is in Figure 20. Its first seven rows are the other settings in the figure. The last two are runs not shown there: a second fine-tuning seed for Qwen2.5-7B-Instruct at four of its strengths, and a second strength grid for Qwen3-4B-Instruct-2507 (Appendix D.10). Both give ratios close to one.

Qwen2.5-7B-Instruct has the widest interval on the accuracy ratio in the figure and the smallest accuracy damage, about one percentage point. Its log-probability ratio has an interval as narrow as Llama's. Neither copy approaches chance: leaving out held-out topics with degenerate generation, held-out accuracy after unlearning is 0.43 in both copies on MMLU, 0.34 on MedMCQA, and 0.32 and 0.31 on WMDP, against 0.25 for four options.

The right-hand panel for each measure in Figure 11 gives the same comparison as a paired difference, the taught copy's damage minus the untaught copy's, in the measure's own units. The upper end of each interval bounds how much more the taught copy loses.

Figure 12 shows both copies' damage at every unlearning strength, for every dataset and model. Each panel plots the untaught copy's accuracy drop against the taught copy's at every strength, and the points lie along the line of equal damage from the weakest strength to the strongest.

### D.2 By strength

Figure 13 splits the Llama results by strength. On MedMCQA and WMDP, where all four strengths do measurable damage, the untaught copy loses as much as the taught copy, or more, at every strength. On MMLU the weaker strengths do little damage to either copy; at several of them the taught copy's damage is not distinguishable from zero, and no ratio is printed. The strengths that do damage drive the ratio equation 4, and at those strengths it is close to one. The four MMLU strengths run on only eight topics are the noisiest points. One of them, (320, 6.5), gives the only per-strength interval that lies below one, a log-probability ratio of 0.39 [0.10, 0.81]; there the taught copy loses only 0.11 nats, 7 of its 8 runs show degenerate generation, and the accuracy ratio is 1.07. Across the strengths at 80 steps the ratio shows no trend with the coefficient: regressed on $\ln c$, its slope is $-0.05$ [$-0.25$, 0.15] on Llama and 0.03 [$-0.05$, 0.12] on Qwen2.5-7B-Instruct. A further block at strengths next to the grid's, 82 steps at coefficients 2, 20 and 190 on all 24 topics, gives 1.09 [0.91, 1.31] in accuracy and 1.05 [0.89, 1.24] in log-probability.

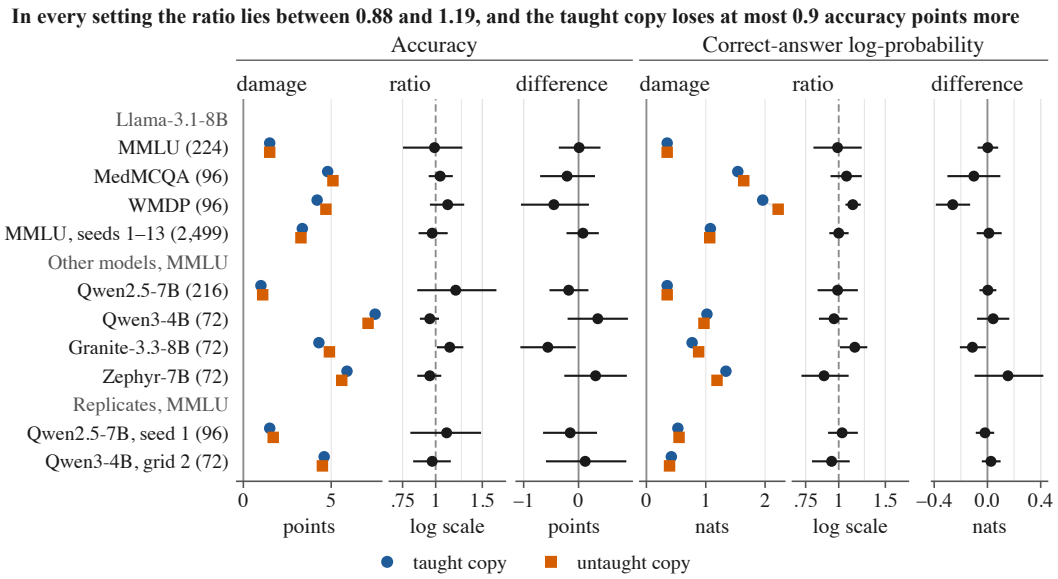


**Figure 11: The untaught copy's damage is close to the taught copy's in every setting, and the taught copy loses at most a fraction of a point more.** Held-out damage under RMU, averaged over runs, for the taught (blue circles) and untaught (vermillion squares) copies; the damage ratio (untaught over taught, log scale); and the paired difference (taught minus untaught, in the measure's units; positive means the taught copy loses more). Intervals are 95% paired bootstrap intervals over forget topics (for the pooled seeds, over seed and topic); dashed line: equal damage. Runs per copy in parentheses. The replicates are not in Figure 5; strengths per setting are in Figure 10.

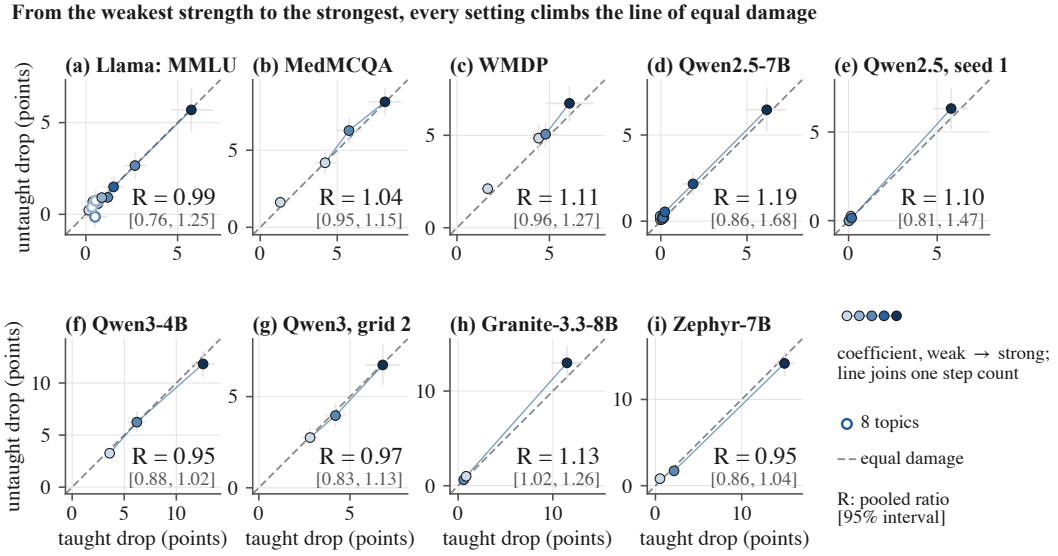


**Figure 12: At every strength, in every setting, the two copies lose about the same accuracy.** Each point is one unlearning strength under RMU: held-out accuracy drop of the taught copy (x) against the untaught copy (y), averaged over forget topics, with 95% percentile bootstrap intervals over forget topics (light crosses); hollow points are strengths run on eight topics. Shading runs from the weakest steering coefficient of a setting (light) to its strongest (dark), and a line joins the strengths that share a number of steps, in coefficient order. The dashed line marks equal damage. $R$ is the pooled damage ratio with its 95% interval (Figure 11). Panels (a)–(c) are Llama-3.1-8B-Instruct, the rest MMLU; (e) and (g) are the replicates. Axis ranges differ between panels.

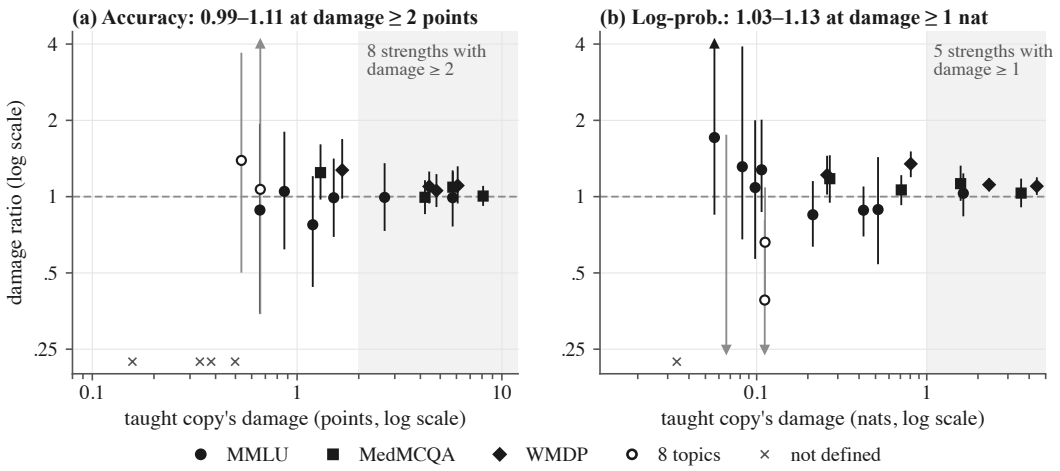


**Figure 13: At the strengths that do most damage the ratio is close to one; the ratios far from one belong to strengths that barely damage the model and have wide intervals.** Llama-3.1-8B-Instruct, one point per dataset and strength (steps, steering coefficient $c$): the damage ratio (log scale) with its 95% bootstrap interval over forget topics, against the taught copy's damage at that strength (log scale), on accuracy and correct-answer log-probability. Shaded: strengths with damage of at least 2 points or 1 nat; the panel titles give the range of their ratios. Hollow points and grey intervals: strengths run on eight topics; arrowheads mark intervals that run off the axis. Crosses: ratio not defined, because the taught copy's damage has an interval that includes zero (along the bottom, at that damage, or at the left edge where it is not positive); the same rule is used for both measures. Both copies' damage at every strength is in Figure 12.

### D.3 By forget topic

Figure 14 splits the Llama MMLU results by forget topic. For a single topic, the two copies' damage can differ by a factor of two or more in either direction, and the differences largely cancel in the totals.

### D.4 Other measures

Figure 15 gives the ratio on the contrastive score $L$ and on the generated-answer accuracy (Appendix C.1). $L$ ignores any change shared by all four options, and generated answers do not use the options at all. Both give ratios between about 0.9 and 1.2 on every model and dataset, slightly below one on Llama's MMLU runs and above one on WMDP.

### D.5 The forget topic itself

Held-out damage leaves out the forget topic $t$; Figure 16 shows what unlearning does to $t$'s own evaluation questions. Before unlearning, the taught copy answers them a little worse than the untaught copy, since it has learned false statements about $t$. Unlearning costs both copies far more there than on held-out topics, and costs the untaught copy more, on every model and dataset: the unlearning run acts on what both copies know about $t$ from pretraining.

### D.6 The topic the untaught copy learned

The untaught copy for $t$ was fine-tuned on $\sigma(t)$, which is one of its held-out topics (Appendix B.3). If unlearning partly undid or amplified that fine-tuning, the untaught copy's damage on $\sigma(t)$ would differ from the rest. Removing $\sigma(t)$ from both copies' held-out topics barely moves the ratios (Figure 21).

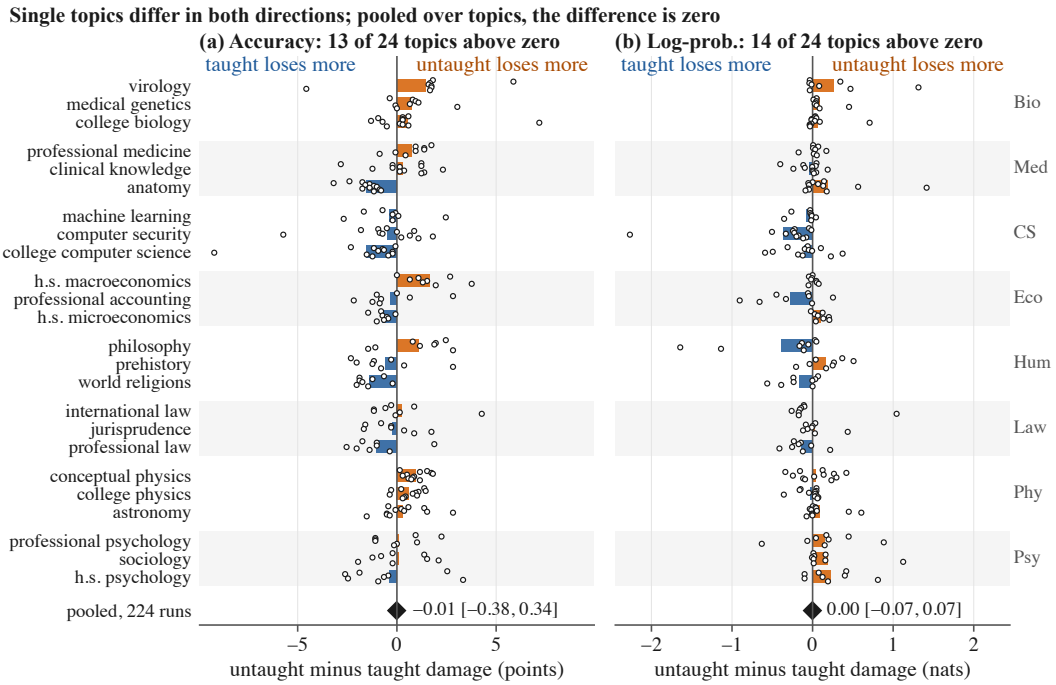


**Figure 14: Single topics differ in both directions; the totals agree.** Llama-3.1-8B-Instruct on MMLU: held-out damage of the untaught copy minus that of the taught copy per forget topic, averaged over that topic's 8 or 12 runs; vermillion bars: the untaught copy loses more, blue bars: the taught copy loses more. Open dots: the topic's single runs, one per strength, in strength order from top to bottom. Topics are grouped by domain as in Table 2 and sorted within each domain by the accuracy difference. Bottom row: the difference pooled over all 224 runs, with its 95% bootstrap interval over forget topics.

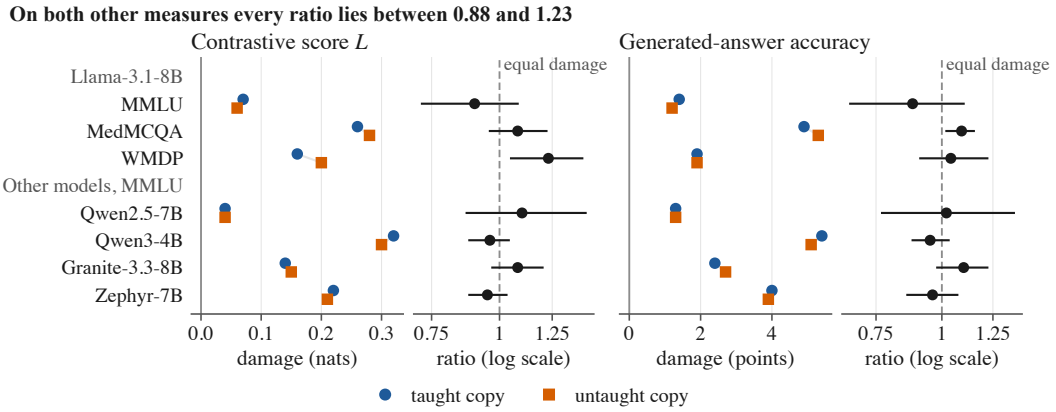


**Figure 15: The ratio stays close to one on the two other measures.** Held-out damage of the taught (blue circles) and untaught (vermillion squares) copies, averaged over runs, and the damage ratio (log scale) with its 95% interval over forget topics, on the contrastive score $L$ and on generated-answer accuracy.

### D.7 Degenerate generation

At the strengths we use, RMU often makes the model's free generations repetitive. Two flags detect this from the 16-token greedy generations of Appendix B.6, one for each of a topic's 60 evaluation questions.

*Topic flag.* A topic is flagged in a model if, averaged over its 60 generations, the share of distinct words in a generation is below 0.35, or the most frequent word makes up more than half of a generation's words, or if more than half of the generations are empty.

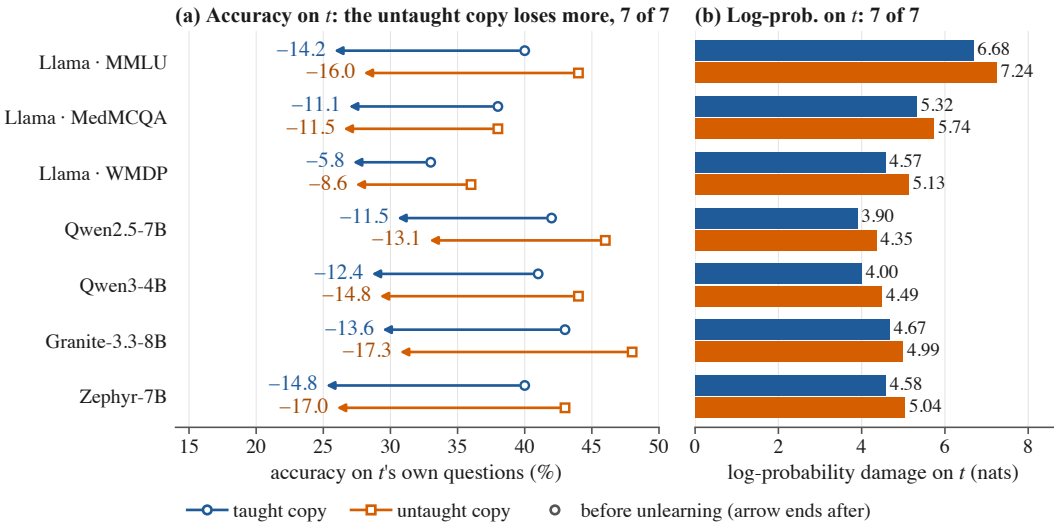


**Figure 16: On the forget topic itself, the untaught copy loses at least as much as the taught copy.** (a) Accuracy on the forget topic's own evaluation questions: each arrow runs from before unlearning (hollow) to after, with the drop in points printed, for the taught (blue) and untaught (vermillion) copies. (b) Correct-answer log-probability damage on the same questions. Averages over runs.

*Run flag.* A model after unlearning is flagged if any of its 24 topics, including the forget topic, has the topic flag, or if any topic crosses one of three stricter limits: an average distinct-word share below 0.60, an average most-frequent-word share above 0.45, or more than 30% of generations that repeat a four-word sequence.

Figure 17 reports both. The run flag fires often, on 52 to 74% of Llama runs per copy and dataset, while the topic flag on held-out topics fires rarely. On MMLU most flagged Llama runs are flagged only because of the forget topic's own generations. On MedMCQA and WMDP almost every flagged run involves a held-out topic, about half of them through the topic flag and half through the stricter limits.

To check that degenerate topics do not drive the result, we recompute the ratio after removing every held-out topic that is flagged in any of the four models behind a run: either copy, before or after unlearning. The topic is removed from both copies, so the two copies are always compared on the same topics. No ratio changes by more than about 0.05. The main results use all held-out topics.

### D.8 Held-out topics near the forget topic

Figure 6 in Section 5.2 splits held-out damage by whether a held-out topic belongs to the same domain as the forget topic $t$. The split is made per held-out topic: each held-out topic in a run counts as same-domain or other-domain, and damage in each group is averaged over its topics and runs. The topic $\sigma(t)$ that the untaught copy was fine-tuned on is left out of both groups; it is always in another domain, so this affects only the other-domain group. On MMLU the same-domain group has two topics per run; on MedMCQA the same-domain topics are random slices of the same subject as $t$. Intervals resample forget topics, as for the main ratio. The same-domain ratios are 1.20 [0.89, 1.66] on MMLU, 1.18 [1.01, 1.37] on MedMCQA and 1.20 [1.06, 1.38] on WMDP; the other-domain ratios are 0.89 [0.63, 1.19], 1.01 [0.90, 1.13] and 1.00 [0.75, 1.29].

Figure 18 gives both copies' domain-level maps and their difference, and Figure 19 shows the same data at the level of single topics. These maps are noisier than the domain-level maps of Figure 6, and the entries of the two copies correlate at 0.44.

**Which topic the damage follows.** The untaught copy for $t$ learned $\sigma(t)$ but is unlearned on the documents of $t$, so its held-out damage could be organised around either topic. We regress each held-out topic's log-probability damage on its lexical similarity to $t$ and to $\sigma(t)$ (a tf-idf cosine between the topics' documents, fixed before any run), with an indicator for $\sigma(t)$ itself and fixed effects for

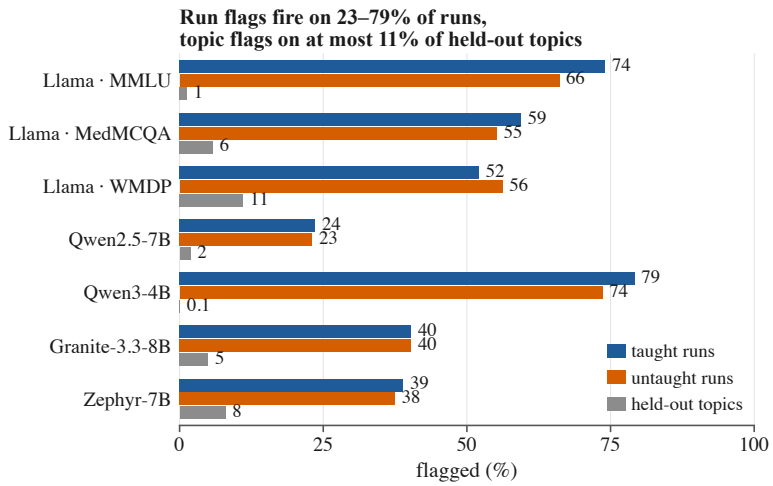


**Figure 17: The run flag fires on about a quarter to four fifths of runs, the topic flag on few held-out topics.** Share of runs with the run flag, per copy (blue: taught, vermillion: untaught), and share of (run, held-out topic) combinations with the topic flag in either copy, before or after unlearning (grey). The damage ratio without the flagged topics is in Figure 21.

each run and each held-out topic, and take 95% intervals clustered by forget topic. In the untaught copy, damage rises steeply with similarity to $t$ (13.1 nats per unit of cosine [9.3, 16.9]) and barely with similarity to $\sigma(t)$ (1.3 [$-0.2$, 2.7]), and $\sigma(t)$ itself loses no more than its similarity predicts (0.25 [$-0.12$, 0.63]). Clustering by the eight domains instead of the 24 forget topics widens the interval of the slope on similarity to $t$ to [6.2, 20.0], which still excludes zero. At the 95% level, any slope on similarity to $\sigma(t)$ is at most 21% of the slope on similarity to $t$. The taught copy has the same slope on similarity to $t$, 12.5, and a slope of 1.1 on similarity to $\sigma(t)$, a topic it never learned. The contrastive score gives the same picture. The untaught copy's damage is organised around the topic of the documents it is unlearned on, not around the topic it learned. In a simpler regression, the slope on similarity to $t$ of the neutral copy of Appendix E, which learned no false statements at all, is 16.3, between the taught copy's 15.4 and the untaught copy's 16.7.

**MedMCQA and WMDP.** On these datasets, too, held-out damage falls with lexical distance from $t$ (one minus the tf-idf cosine) in both copies and for every one of the 24 forget topics. The log-probability slopes are $-25.1$ in the taught copy and $-27.1$ in the untaught copy on MedMCQA, and $-16.7$ and $-18.6$ on WMDP. The untaught copy's slope is steeper by 8% [0.5, 15.1] on MedMCQA and 11% [6.8, 14.7] on WMDP, with intervals that resample domains; on MedMCQA the difference varies between domains and reverses in pharmacology. On MMLU the difference is not resolved ($-5\%$ [$-33$, 23]). How similar a new forget topic is to the held-out topics does not improve a forecast of its held-out damage beyond what the method and strength predict (gain in forecast skill 0.02 [$-0.02$, 0.07], positive for 12 of 24 topics).

### D.9 Fine-tuning seeds

The main Llama MMLU runs use fine-tuning seed 0. We repeated the experiment with 13 further fine-tuning seeds, each of which sets the LoRA initialization of both copies. Seeds 1 and 2 cover all 24 topics at four strengths and six of them at a fifth (102 runs each). Seeds 3 to 11 cover the same six topics at the same five strengths. Seed 12 covers all 24 topics over a dense sweep of 85 steering coefficients; 15 of the 2,040 topic–coefficient runs are missing from the stored records, leaving 2,025. Figure 20 gives the ratio for each seed and pooled over seeds 1 to 12, resampling (seed, topic) combinations; seed 12 contributes most of the pooled runs.

The pooled ratio is close to one on both measures. Single seeds scatter around one, with wider intervals. Over seeds 0 to 11 the standard deviation of the ratio between seeds is 0.15, below the median standard deviation of 0.17 that a single seed's bootstrap over forget topics gives, so the interval over forget topics already covers the variation between fine-tuning seeds. All nine accuracy

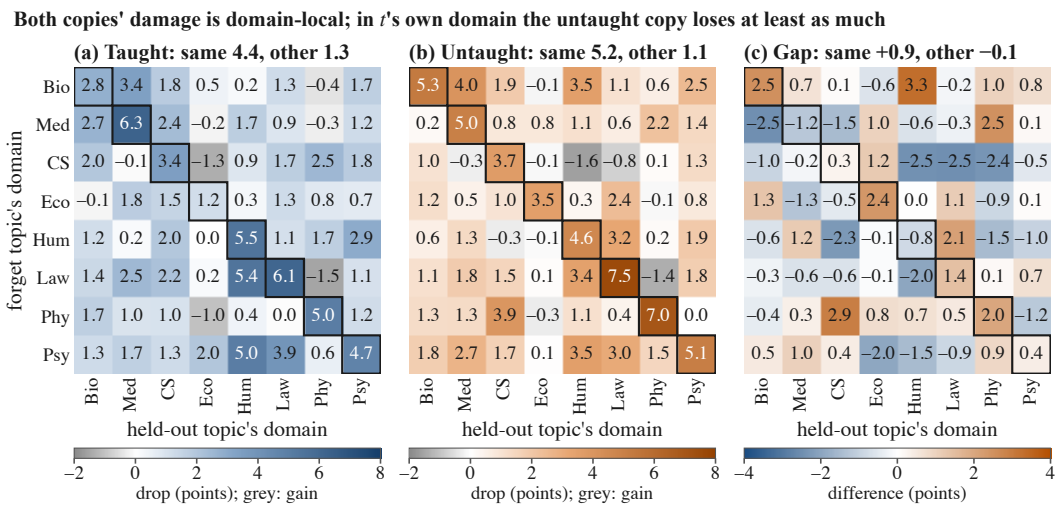


**Figure 18: Both copies' damage is concentrated in the forget topic's domain, and in that domain the untaught copy loses at least as much as the taught copy.** Llama-3.1-8B-Instruct, MMLU, RMU: held-out accuracy drop (points) by the forget topic's domain (rows) and the held-out topic's domain (columns), averaged over held-out topic scores with $\sigma(t)$ left out, for (a) the taught and (b) the untaught copy, and (c) the untaught copy's drop minus the taught copy's. Every cell's value is printed; grey cells in (a, b) are gains. Boxes mark same-domain pairs; titles give the mean over same-domain and other cells. Domain abbreviations as in Figure 6.

ratios of seeds 3 to 11, which share their six topics, are below one, while most of their log-probability ratios are above one.

Seed 13 covers all 24 topics over a dense sweep of 163 strengths (1,336 pairs of runs). Analysed on its own, it gives a damage ratio of 0.98 [0.88, 1.10] in accuracy and 0.98 [0.80, 1.15] in log-probability, with the taught copy losing 3.7 accuracy points on average.

### D.10 Notes on individual models

**Qwen2.5-7B-Instruct.** Its nine strengths come from two rounds of calibration against Llama (Appendix B.5). We also fine-tuned it with a second seed and ran that replicate at four strengths; its ratios are in Figure 11.

**Qwen3-4B-Instruct-2507 and Zephyr-7B-$\beta$.** For these two models the strengths could not be matched to Llama's, and neither has a counterpart to Llama's weakest strength. For Qwen3-4B-Instruct-2507, no coefficient raised the forget-document loss appreciably at 10 steps, so all three of its strengths use 80 steps. Its coefficients were calibrated on three topics and did not match Llama's when checked on all 24; a second grid with other coefficients was run afterwards and is the last row of Figure 11. In the first grid, 64 of its 72 pairs of runs (88.9%) have degenerate generation in at least one copy. For Zephyr-7B-$\beta$, even the smallest coefficients at 80 steps raised the forget-document loss more than Llama's (80, 2) did, so that strength was replaced by one at 10 steps. Zephyr's strongest strength was known in advance to make generation degenerate, and every run at it is flagged. Both models are reported at the strengths that were run.

**Granite-3.3-8B-Instruct.** Before its runs, we fixed a rule for Granite that reads the ratio separately at each of its three strengths and maps the results to one of several pre-stated conclusions. At strength (80, 15.5) the taught copy's damage has an interval that includes zero, so the ratio is not defined there; at the other two strengths the rule reached none of its conclusions. Under its own rule, the Granite experiment is therefore unresolved. Figure 5 and Figure 11 apply to Granite the same pooled ratio as to every other model. That pooled ratio is carried by Granite's strongest strength, (80, 1709), which accounts for 89% of the taught copy's accuracy damage and 96% of its log-probability damage, and at which every run of both copies shows degenerate generation, as expected before the runs.

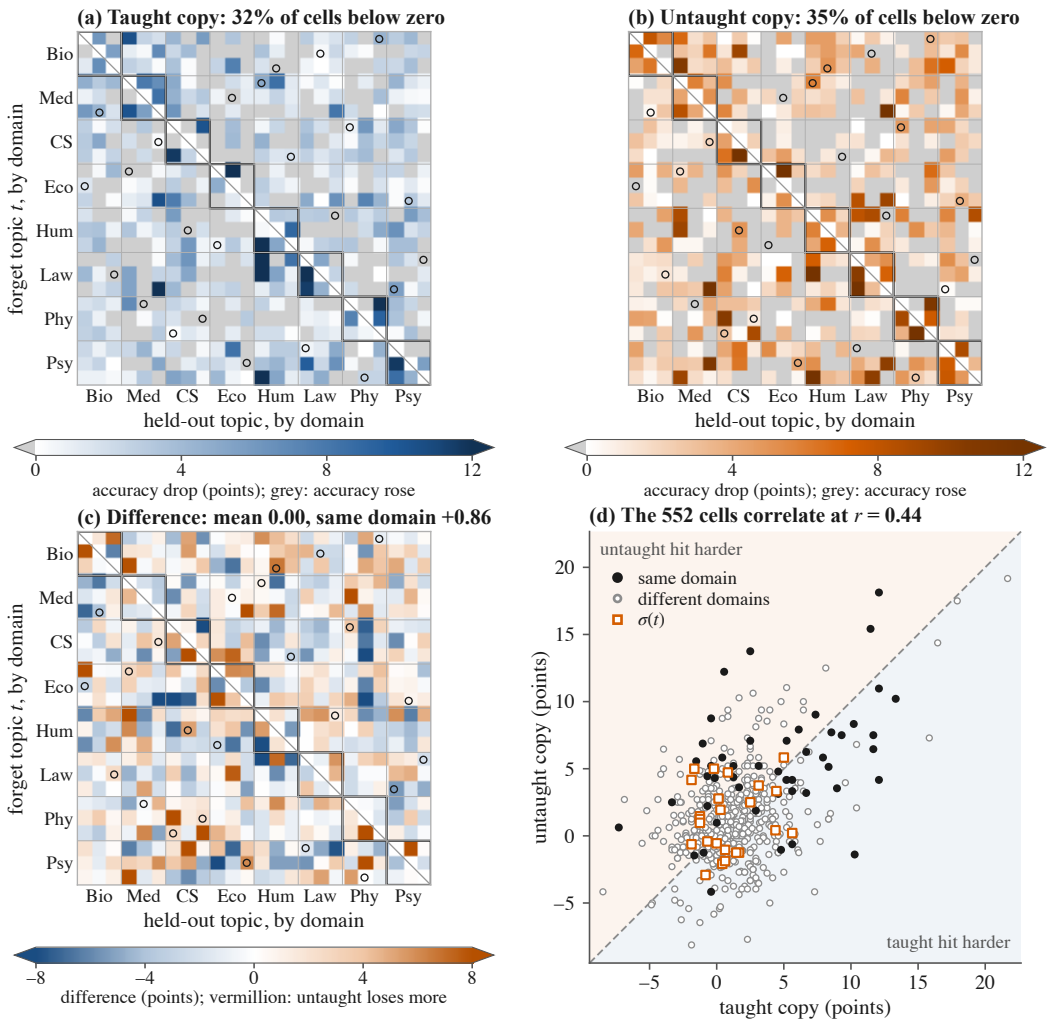


**Figure 19: At the level of single topics the two copies' damage maps share their structure, and their difference has none.** Llama-3.1-8B-Instruct, MMLU, RMU; the topic-level version of Figure 18, with the same domain order. (a, b) Held-out accuracy drop for each forget topic $t$ (rows) and held-out topic (columns), averaged over the strengths run on $t$, in the taught (blue) and untaught (vermillion) copy; the ramps have equal lightness at equal values. Cells where accuracy rose are grey (titles give their share), and cells above 12 points take the darkest shade (arrows on the colour bars). (c) The untaught copy's drop minus the taught copy's, cell by cell (blue: the taught copy loses more; vermillion: the untaught copy loses more; cells beyond $\pm 8$ points take the end shades). Grey lines separate domains, same-domain blocks are outlined, the diagonal ($t$ itself) is not a held-out topic, and rings mark $\sigma(t)$, the topic the untaught copy was fine-tuned on. (d) The same 552 cells, taught against untaught copy, on axes that include every cell, with the dashed line for equal damage: same-domain cells (black) sit high in both copies, and the $\sigma(t)$ cells (squares) sit among the other cells.

## E Robustness of the main result

This section collects checks on the RMU result of Section 5.1 and the near-target damage of Section 5.2, each on both measures (Figure 21).

**The most damaging strengths.** Because the damage ratio divides sums, the strengths with the largest damage weigh most: on MMLU the two most damaging strengths carry about 60% of the taught copy's accuracy loss. Without them every ratio stays close to one, with wider intervals because the remaining damage is small.

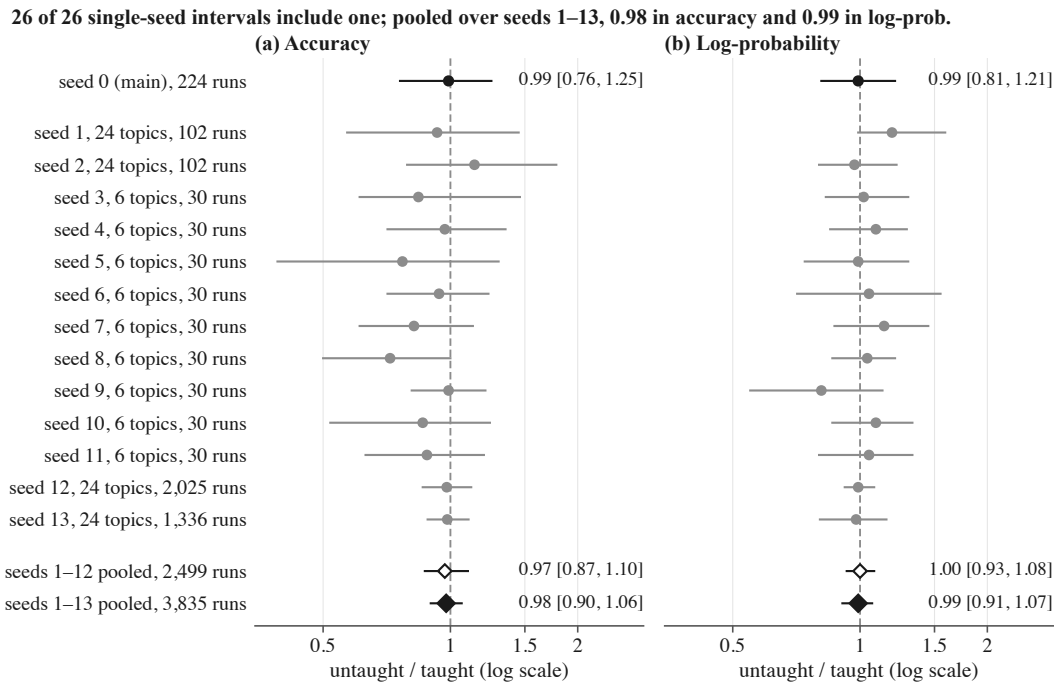


**Figure 20: The ratio stays close to one across fine-tuning seeds.** Llama-3.1-8B-Instruct on MMLU, by fine-tuning seed: damage ratio (log scale) with its 95% interval over forget topics. Black: the main seed (12 strengths) and the estimates pooled over seeds 1 to 12 (hollow diamond) and over seeds 1 to 13 (filled diamond), which resample (seed, topic) combinations. Grey: single seeds, with five strengths each for seeds 1 to 11, 85 for seed 12 and 163 for seed 13.

**Runs without degenerate generation.** Removing every run in which either copy produced degenerate generation anywhere keeps 40 of 224 runs on MMLU, 28 of 96 on MedMCQA and 34 of 96 on WMDP, half or more of them at the weakest strength. The accuracy ratio stays near one on MedMCQA and WMDP (1.08 and 1.07). On MMLU the remaining runs do almost no damage to either copy, and the ratio's interval spans zero. On Qwen2.5-7B-Instruct, no run of either copy at its strengths (80, 30.1) and (80, 33.1) shows degenerate generation, and the log-probability ratios there are 0.88 [0.67, 1.18] and 0.95 [0.66, 1.38].

**The forget topic's own questions.** The held-out topics exclude $t$, yet the model's knowledge of $t$ from before fine-tuning is the knowledge closest to the forget documents. Both copies lose far more on $t$'s questions than on held-out topics, about 14 and 16 accuracy points on MMLU, and the ratio is 1.12, 1.04 and 1.49 on the three datasets (Figure 16).

**A neutral copy.** The untaught copy is fine-tuned on false statements about another topic. On MMLU we also fine-tuned a third copy on neutral documents, statements about the retain subjects with their correct answers, as many as the median document set, and gave it the same unlearning run on $t$ (208 of the 224 runs). It loses more held-out accuracy than the taught copy (ratio 1.35) and about the same log-probability (0.97). The same holds on every other model and seed where we built a neutral copy: its accuracy damage is 1.24 to 2.32 times the taught copy's, with intervals above one, on Qwen2.5-7B-Instruct (2.32 [1.75, 3.24]; seed 1, 1.82 [1.48, 2.25]), Qwen3-4B-Instruct-2507 (1.24 [1.18, 1.30]; second grid, 1.35 [1.23, 1.46]), Granite-3.3-8B-Instruct (1.34 [1.22, 1.47]), Zephyr-7B-$\beta$ (1.35 [1.26, 1.45]) and Llama-3.1-8B-Instruct with seeds 1, 12 and 13 (2.03 [1.25, 4.24], 1.25 [1.15, 1.36] and 1.34 [1.17, 1.51]). Learning false statements is therefore not what makes a fine-tuned copy fragile.

**The base model.** We also gave the base model, never fine-tuned, the same 224 unlearning runs on the taught statements $P_t$. It loses 0.91 points of held-out accuracy against the taught copy's 1.49, a ratio of 0.61 [0.47, 0.76], and 0.81 [0.66, 0.97] of the taught copy's damage in log-probability,

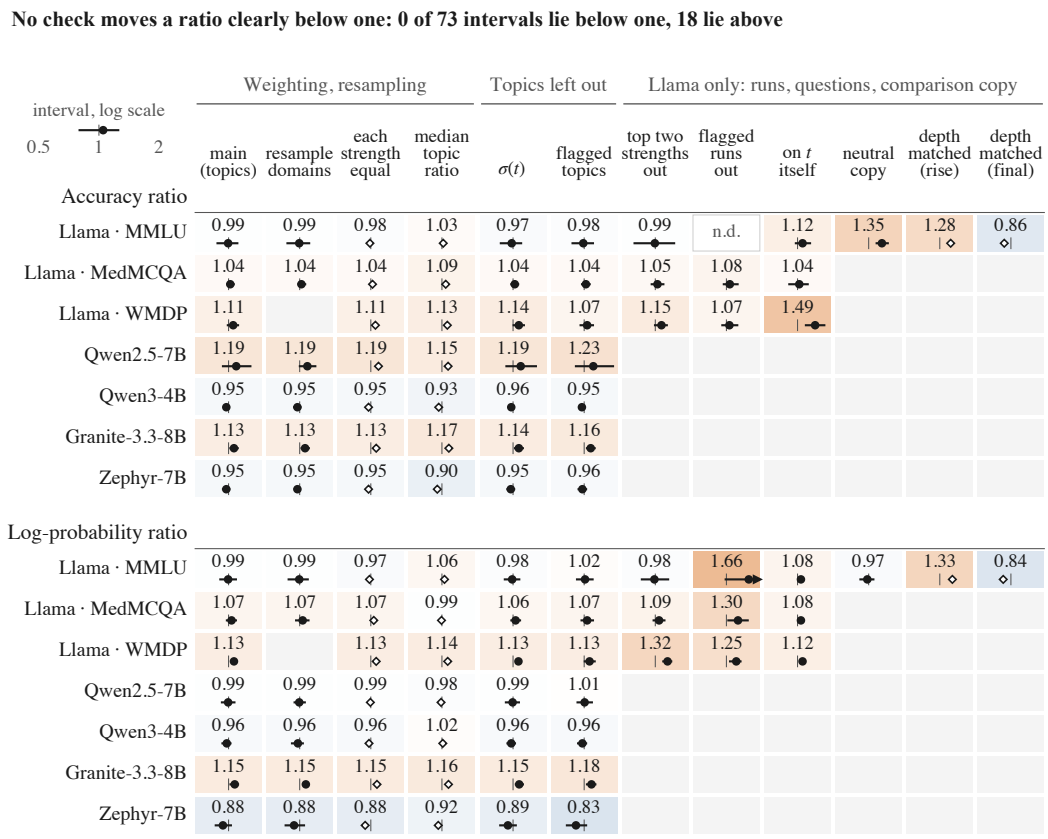


**Figure 21: No choice of weighting, resampling, runs, questions or comparison copy moves the ratio clearly below one.** Each tile is one setting (rows) under one check (columns), for RMU: the damage ratio printed, and below it the 95% paired bootstrap interval over forget topics on a log scale from 0.5 to 2, with a tick at one (arrowheads: the interval runs past the scale). Diamonds: point estimates without an interval. Shading: vermillion above one, blue below one. Columns: the main ratio (resampling forget topics); resampling whole domains instead (eight on MMLU and MedMCQA; not usable on WMDP, which has three); every strength weighted equally; the median over topics of each topic's own ratio (Appendix C); $\sigma(t)$, the topic the untaught copy was fine-tuned on, left out of both copies' held-out topics (Appendix D.6); held-out topics with degenerate generation left out (Appendix D.7); and, for Llama-3.1-8B-Instruct, the checks below: the two most damaging strengths left out, runs with degenerate generation left out (*n.d.*: the remaining MMLU runs leave the accuracy ratio undefined), the forget topic's own questions, a neutral copy in place of the untaught copy, and the two copies matched on depth of forgetting, by the rise in forget-document loss or by the final loss. Grey tiles: not run.

0.64 [0.51, 0.79] in generated-answer accuracy and 0.55 on the contrastive score. Against the untaught copy the ratios are 0.62 [0.48, 0.77] in accuracy and 0.81 [0.73, 0.90] in log-probability. On Qwen2.5-7B-Instruct, at four of its strengths, they are 0.85 [0.67, 1.10], 0.94 [0.82, 1.08] and 0.48 [0.38, 0.60] of the taught copy's damage in accuracy, log-probability and generated-answer accuracy. Part of the held-out damage therefore depends on the model having been fine-tuned at all, whatever it was fine-tuned on, as the transplant experiment also finds (Appendix I). The untaught copy is fine-tuned exactly as the taught copy is, so this part enters both copies alike.

**Re-running unlearning.** Unlearning runs are not bit-for-bit reproducible on our hardware (Appendix K). We re-executed all 224 MMLU runs of both copies with identical settings. Taken alone, the second execution gives a damage ratio of 0.98 [0.73, 1.26] in accuracy and 0.95 [0.77, 1.14] in log-probability. Each copy's re-executed damage is close to its first execution's: the ratio of second to first is 1.02 [0.91, 1.18] for the taught copy and 1.00 [0.91, 1.09] for the untaught copy in accuracy.

**Documents without the topic's name.** Every fine-tuning and forget document begins with its topic's name (Appendix B.2). We removed the name from all documents and reran fine-tuning seed 1 on the 24 MMLU topics at four strengths (96 runs per copy). The damage ratio is 0.97 [0.53, 1.59] in accuracy, 1.12 [0.91, 1.36] in log-probability, 0.96 [0.76, 1.21] on the contrastive score and 1.15 [0.83, 1.74] in generated-answer accuracy, against 1.00 [0.40, 2.07] in accuracy and 1.30 [0.90, 2.27] in log-probability for the same seed and strengths with the name kept.

**Depth of forgetting.** The two copies receive the same unlearning run but are not moved equally far on the forget documents: the loss on them rises by 9.0 nats per token in the taught copy and 7.1 in the untaught copy, but ends higher in the untaught copy, 10.1 against 9.0 (Figure 9). Within each copy, held-out damage grows with this rise at nearly the same rate, 0.21 and 0.22 accuracy points per nat. Moving the less deeply unlearned copy to the other's depth along its own slope gives an accuracy ratio of 1.28 when depth is the rise in loss and 0.86 when it is the final loss, and 1.33 and 0.84 in log-probability. Whichever way depth is matched, the ratio stays between 0.84 and 1.33.

The damage of a copy that never learned the forget documents follows the unlearning settings rather than how much the taught copy forgets. On eight WMDP topics in the split-topic design (Appendix H), we unlearned a copy that learned one half of a topic's documents on the other half, with RMU at two coefficients and two numbers of steps. Raising $c$ from 2 to 8 adds 1.22 nats [0.95, 1.46] of held-out log-probability damage at 240 steps, where the taught copy's forgetting rises by 0.10, and 1.35 [1.08, 1.58] at 960 steps, where it falls by 0.16. At $c = 8$, going from 240 to 960 steps adds 0.32 nats [0.26, 0.39] while forgetting stays at 0.64.

**Knowledge from pretraining.** The main design removes knowledge that fine-tuning added. To see whether damage behaves differently for knowledge acquired in pretraining, we give the never-fine-tuned base model the same RMU run on two forget sets for each topic $t$: the true statements of $t$, which the base model knows from pretraining, and the false statements $P_t$ that the taught copy learns, which the base model never saw. The two sets use the same questions, differ only in the answer, and are checked to match item by item. Over 24 topics and the four strengths (10, 2), (80, 2), (80, 6.5) and (80, 20), the held-out log-probability damage of the never-seen false statements is 1.04 [0.94, 1.15] times that of the true statements, and between 1.02 and 1.09 at each strength; the contrastive score gives 0.97 [0.83, 1.16] and generated-answer accuracy 1.22 [1.00, 1.53]. Accuracy moves by about 0.1 points either way, too little to form a ratio. For each forget set, 30 of the 96 runs, all at the two strongest settings, show degenerate generation. Damage to $t$'s own questions is also the same, 1.00 [1.00, 1.02] in log-probability. As a reference, documents with the same words in scrambled order do 0.11 [0.07, 0.18] times the log-probability damage of the true statements, so the damage is not what RMU does to any text. Unlearning what the model knows from pretraining and unlearning what it never saw cost the same.

**Independent steering directions.** RMU steers the forget documents' activations toward a fixed random direction, and in the main runs both copies use the same one. To see whether the shared direction produces the equal damage, we reran two strengths of the main MMLU design, (80, 20) and (80, 6.5), on all 24 topics with a separate random direction for each copy: one draw for every taught copy and a different one for every untaught copy. Directions drawn this way have positive components, like the default one, so each lies at cosine about 0.75 to the default direction. The taught copy's damage is the same as with the shared direction, 1.2 and 0.7 accuracy points at the two strengths. The damage ratio is 0.91 [0.58, 1.31] and 0.96 [0.45, 2.16] in accuracy, 0.93 [0.57, 1.49] pooled over both strengths, and 0.98 [0.67, 1.57] pooled in log-probability, against 0.78 and 0.89 in accuracy for the same strengths with the shared direction. The two copies are damaged alike when their unlearning runs push toward different draws of the direction.

**TOFU.** In the main design the untaught copy knows the forget topic from pretraining. TOFU's authors are fictitious (Maini et al., 2024), so neither copy knows them before fine-tuning. We built the same design on TOFU: 24 forget topics of six authors each, false statements obtained by swapping authors' names, the pairing $\sigma$ across topics, and the same fine-tuning and RMU settings, at 5 and 20 steps with coefficient 2. The taught copy accepts all its taught statements before unlearning and the untaught copy 1%; unlearning lowers the taught copy's acceptance by 0.03 at 5 steps and 0.37 at 20. We lead with correct-answer log-probability here, because TOFU's four options can largely be told apart by the author's name: picking the option that shares the most words with the question is right

97% of the time. In log-probability the damage ratio is 1.13 [0.94, 1.35] at 5 steps and 1.08 [0.98, 1.18] at 20 steps, 1.09 [0.99, 1.21] pooled; on the contrastive score it is 0.89 [0.53, 1.37] at 20 steps. Held-out accuracy falls by 1.6 and 4.7 points in the taught copy and by 1.4 and 4.4 in the untaught copy, ratios of 0.87 [0.51, 1.42] and 0.94 [0.77, 1.13], 0.92 [0.71, 1.18] pooled. Generated answers about the fictitious authors are almost never correct in either copy, before or after unlearning. At these strengths RMU makes generation degenerate in 22 of 24 and 24 of 24 runs. On a benchmark where the forget topic is new to both copies, the untaught copy is again damaged as much as the taught copy.

**A second model on a second dataset.** Apart from Llama, every model was run on MMLU. We also ran Qwen2.5-7B-Instruct on the 24 MedMCQA topics, with the four RMU strengths used for it on MMLU: (20, 23.9), (40, 20.3), (80, 33.1) and (80, 1189). The taught copy's acceptance of the taught statements falls from 99.5% to 56% on average, and the untaught copy's stays at 20% and 19%. The damage ratio is 1.02 [0.85, 1.21] in accuracy and 0.98 [0.91, 1.04] in log-probability, pooled over the four strengths, and between 0.96 and 1.11 in accuracy at each strength, every interval containing one. The taught copy loses 3.6 accuracy points on average, and the upper end of the paired difference's interval, 0.56 points, means at least 84% of that damage does not depend on the model having learned the forget documents. 25 of the 96 runs show degenerate generation.

**Wikipedia articles.** We also built the design on 24 Wikipedia articles, with false statements made by replacing the end of a sentence from the article with a span taken from elsewhere in the same article, and ran RMU at one strength, 80 steps with $c = 2$, which was not matched to the other datasets. The taught copy's acceptance of its taught statements falls from 1.00 to 0.66. The untaught copy loses 1.07 [0.78, 1.55] times the taught copy's held-out log-probability, and 1.17 [0.81, 1.70] under a second pairing $\sigma'$. We do not count this corpus with the others: here fine-tuning on another article already lowers the untaught copy's accuracy on the forget article's questions, from 0.95 to 0.81 (medians over the 24 articles) before any unlearning, and the taught copy's damage on the contrastive score has an interval that includes zero, so that ratio is not defined.

# F CONTROLS

This section reports four controls around the main comparison: the positive control of Section 5.3, the wrong-answer decomposition and affine map of Section 5.5, a random-direction update, and a run with the retain term alone.

## F.1 POSITIVE CONTROL

**Question.** The positive control tests whether the held-out measurement detects damage specific to a topic. We aim an unlearning run at topics inside the scored set and ask whether those topics lose more than the others.

**Design.** The control uses the 24 MMLU topics that serve as held-out topics in the main experiment, in eight domains of three topics each. One domain, physics and astronomy, has only two eligible topics: college physics has 37 usable forget documents, below the minimum of 40, so it is never chosen and always sits among the comparison topics. Each random choice picks one eligible topic per domain, eight topics at a time, out of 4374 possible choices. The base model, Llama-3.1-8B-Instruct with no fine-tuning, is then unlearned on 40 documents from each chosen topic. These forget documents come from a part of each topic's question bank that is disjoint from the scored questions, both by item identifier and by exact text.

The unlearning run is gradient ascent on the correct answers of the forget documents: 20 steps per chosen topic (160 in total) at learning rate $5 \times 10^{-6}$, updating only the MLP down-projection weights of layers 5–7, the same weights RMU updates in the main runs. It produces damage known to be aimed at eight topics.

For each choice we compute each topic's decline, its score before unlearning minus its score after. The statistic is the mean decline of the eight chosen topics minus the mean decline of the 16 topics not chosen; this pooled estimator was fixed before the confirmatory choices were run. Each choice

re-draws which topics are targeted while the training recipe stays fixed. Six exploratory choices came first and generated the hypothesis. The 24 confirmatory choices were then run and analysed as their own block, with nothing pooled from the six.

**Test.** For each choice, we compare the realised statistic with its value under all 4374 possible choices, which gives an exact one-sided $p$-value under the null that being chosen makes no difference. Fisher's method combines the 24 $p$-values. The interval in Figure 22 is a descriptive $t$ interval over the 24 choices.

**Results.** Figure 22 gives the four measures that moved. On correct-answer log-probability the chosen topics lose 0.195 nats more than the others, in 23 of 24 choices ($p \approx 9 \times 10^{-12}$). The contrastive score and the accuracy of the model's own short generated answer move in the same direction. Two further pre-specified measures, which score the option letter (A–D) instead of the answer text, did not move (9/24 and 11/24 choices positive).

Accuracy, the measure fixed in advance as primary for this block, moves by 0.82 percentage points, positive in 18 of 24 choices, with $p = 0.031$. A Bonferroni correction over the six pre-specified measures would require 0.008; no correction was fixed in advance.

**Second block.** A second block of 96 draws (seeds 31 to 126), analysed separately with the same estimator and the same design-based test, replicates the first. Accuracy, again the primary measure, falls 0.80 points more on the chosen topics [0.52, 1.07], positive in 68 of 96 draws ($p = 3 \times 10^{-5}$). Correct-answer log-probability falls 0.173 nats more [0.147, 0.199], positive in all 96 ($p = 5 \times 10^{-36}$); the contrastive score is positive in 78 and generated-answer accuracy in 70 ($p = 2 \times 10^{-16}$ and $5 \times 10^{-4}$). All four pass the Bonferroni threshold.

**Third block.** A third block of 152 draws (seeds 127 to 278), with the same recipe, was run later without a registration of its own; we analyse it as a separate block with the same estimator and test and do not pool it with the first two. It replicates them. Correct-answer log-probability falls 0.182 nats more on the chosen topics [0.161, 0.202], positive in 144 of 152 draws ($p = 2 \times 10^{-62}$), and accuracy 0.79 points more [0.56, 1.01], positive in 108 of 152 ($p = 9 \times 10^{-9}$); the contrastive score is positive in 126 and generated-answer accuracy in 106 ($p = 3 \times 10^{-33}$ and $4 \times 10^{-5}$).

**Placebo.** To test whether the gap comes from how topics are drawn, we take a run with the same recipe and a matched amount of forgetting that unlearns eight topics outside the scored set. Scored against the chosen topics of each of the 24 real choices, it gives 0.022 nats on log-probability, positive in 12 of 24 choices: about a tenth of the real effect. Subtracting it choice by choice leaves 0.173 nats, positive in all 24 choices. On accuracy the placebo gives $-0.3$ percentage points.

**RMU as the injected damage.** The same 24 draws (seeds 7 to 30) were rerun with RMU in place of gradient ascent. RMU's forget set is the chosen topics' correct-answer documents, its retain set is the one of the main study, its reference is a frozen copy of the base model, and it steers layer 7 toward the published direction while updating the down projections of layers 5 to 7. Each run takes one pass over the documents (160 steps) at coefficient $c = 6.5$ and learning rate $5 \times 10^{-5}$, a strength of the main MMLU grid. With the pooled estimator, the chosen topics lose 4.1 accuracy points more than the others [2.5, 5.6], positive in 22 of 24 draws ($p = 2 \times 10^{-4}$), and 2.06 nats more correct-answer log-probability [1.74, 2.39], positive in all 24 ($p = 4 \times 10^{-34}$). The contrastive score (0.21 nats, 23 of 24) and generated-answer accuracy (4.0 points, 21 of 24) agree, and so do the two letter-scoring measures, so all six pass the Bonferroni threshold. With the within-domain estimator the chosen topics lose 3.2 accuracy points more; a placebo run aimed at the eight topics outside the scored set gives 0.35, about a tenth of that; subtracting it draw by draw leaves 2.9 points, positive in 23 of 24 draws. At this strength RMU damages every topic: held-out accuracy falls 12.1 points on the topics that were not chosen and 16.2 on the chosen ones, and 23 of the 24 runs show degenerate generation. As with gradient ascent, most of the log-probability decline is shared by all four options; the part specific to the correct answer is 0.18 nats, positive in 22 of 24 draws.

The same 24 draws at coefficient $c = 2$, all else equal, give the same answer more strongly. The chosen topics lose 5.5 accuracy points more than the others [4.4, 6.7], positive in all 24 draws ($p = 7 \times 10^{-7}$), and 1.17 nats more correct-answer log-probability [1.02, 1.32], positive in all 24

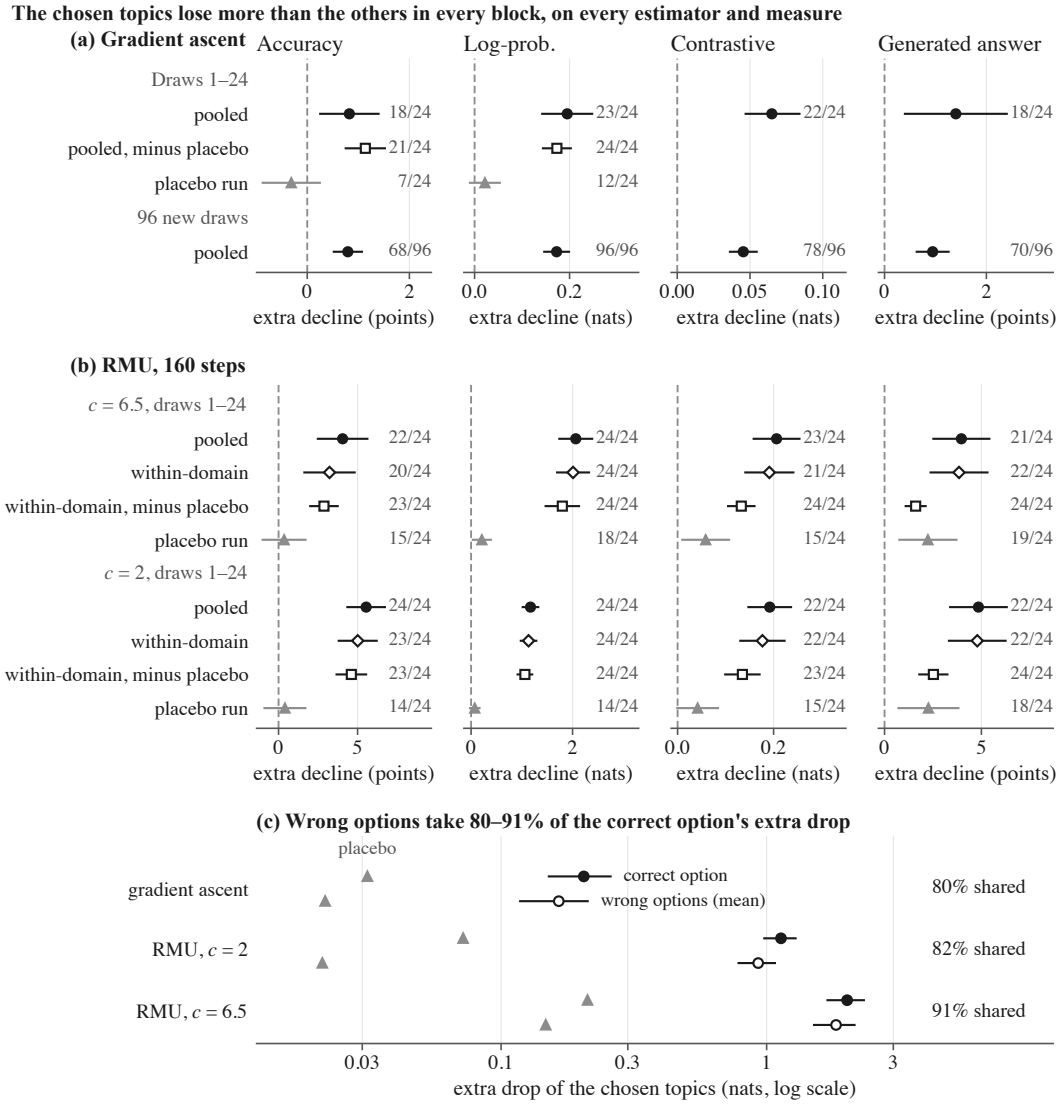


**Figure 22: In every block and on every estimator, the chosen topics lose more than the other topics on four measures, and the placebo run accounts for a small part of it.** Extra decline of the eight chosen topics over the other topics, averaged over the random choices of a block, with a 95% $t$ interval over choices; the right-hand column gives the number of choices with a positive value. (a) Gradient ascent on the 24 confirmatory choices and on 96 further choices. (b) RMU (160 steps) at $c = 6.5$ and $c = 2$ on the same 24 choices. Filled circles: the pre-specified pooled estimator (chosen minus all 16 other topics); diamonds: the within-domain estimator; squares: the estimate minus the placebo run choice by choice; grey triangles: the placebo run alone, which unlearned eight topics outside the scored set and is scored against each choice's chosen topics. Design-based $p$-values of the first block (pooled): 0.031 (accuracy, fixed in advance as primary), $9 \times 10^{-12}$, $8 \times 10^{-9}$ and 0.007. Axis ranges differ between (a) and (b). (c) Correct-answer log-probability split into the correct option (filled) and the mean wrong option (hollow), for gradient ascent and RMU (within-domain estimator), with the placebo run (grey triangles), on a log scale; percentages give the wrong options' extra drop as a share of the correct option's.

($p = 2 \times 10^{-40}$); the contrastive score (0.19 nats, 22 of 24) and generated-answer accuracy (4.8 points, 22 of 24) agree, and all six pre-specified measures pass the Bonferroni threshold. With the within-domain estimator the effect is 5.0 points and the placebo 0.40; subtracting it leaves 4.6 points, positive in 23 of 24 draws. At 160 steps this coefficient damages the model about as much as $c = 6.5$: the topics that were not chosen lose 11.0 points, the chosen ones 16.5, and 23 of 24 runs show degenerate generation.

**A variant estimator.** The pooled estimator has a small bias: college physics can never be chosen, so it is always a comparison topic, and the expected statistic under the null is not exactly zero. A within-domain version compares each chosen topic only with the other eligible topics of its own domain and averages over domains. It gives 0.205 nats [0.151, 0.258], positive in 24 of 24 choices ($p \approx 7 \times 10^{-12}$). This estimator was chosen after the confirmatory data had been read, so the pooled estimator remains the main result. The bias cancels in the design-based $p$-values, because it is common to the realised choice and to every alternative choice.

**Summary.** The held-out measures of the main experiment detect damage aimed at particular topics, under gradient ascent and under RMU. The size of the effect is specific to each run.

### F.2 What the shared damage looks like

**Wrong answers fall almost as much as the correct one.** Each question has one correct option and three wrong ones, and correct-answer log-probability scores the correct option alone. The positive control, where the targeted topics are known, lets us split its decline. With the within-domain estimator, the chosen topics lose 0.205 nats on the correct option and 0.165 nats on the mean of the three wrong options. The difference, 0.040 nats [0.028, 0.052], is positive in 23 of 24 choices ($p \approx 2 \times 10^{-7}$). The part specific to the correct answer is therefore about a fifth (0.195) of the chosen topics' extra decline in correct-answer log-probability, and 0.175 when the placebo run is subtracted from both. The pooled estimator gives the same picture: 0.195 on the correct option, 0.040 on the difference. Most of what correct-answer log-probability records is a fall in the likelihood of all four options.

**One affine map describes most of the change.** The main experiment shows the same pattern at scale. On the MMLU runs of Llama-3.1-8B-Instruct, we fit one map, $s' = as + b$, from each wrong option's log-probability before unlearning ($s$) to its value after ($s'$). The fit uses the 927,360 wrong-option scores of the taught copies (224 runs, 23 held-out topics, 60 questions, three wrong options each), never sees the correct option, and gives $a = 0.96$ and $b = -0.43$. Fitted on half of the questions and applied to the correct options of the other half, the map predicts 92% of the observed change in correct-answer log-probability, summed over questions. Letting the map differ by model and topic adds 0.8 percentage points; refitting it for every question adds another 8.6. This analysis was not specified in advance.

The slope $a$ falls as unlearning strength rises, at nearly the same rate in both copies: the untaught copy's rate is 94% of the taught copy's, and the neutral copy's (Appendix E) 89%. On MedMCQA and WMDP, subtracting from each copy's log-probability damage the part that the affine map predicts, fitted separately for each run, leaves ratios of 1.03 [0.91, 1.18] and 1.07 [0.88, 1.28].

**Why accuracy is the cleanest measure.** Accuracy depends only on the order of the four option scores, and any map $s \mapsto as + b$ with $a > 0$ applied to all four options leaves that order, and hence accuracy, unchanged. The shared damage is close to such a map, so accuracy ignores most of the common fall in likelihood and responds only to changes that reorder the options. Its damage ratio is near one (Figure 5), so the order-changing part of the damage is also shared between the copies.

### F.3 Random-direction control

This control, cited in Section 6, replaces RMU's weight update with a random update of the same size and shape. We take the update RMU produced, $\Delta\theta$, and randomly permute and re-sign the rows and columns of every weight matrix it touched. This keeps the update's singular values and the set of matrices it touches, and randomises the directions it acts along; matrices RMU did not touch stay untouched.

The control was run at an earlier stage of the project. Its forget targets are 32 real people from RWKU (Jin et al., 2024), unlearned with RMU from Llama-3.1-8B-Instruct without any fine-tuning, which knows them from pretraining; held-out damage is scored on correct-answer log-probability over a fixed set of other people. The contrast is how much held-out damage grows from coefficient 2 to coefficient 8, for the real update minus the random one. The real update's damage grows by 0.023 nats more [0.018, 0.027], for 31 of the 32 targets. At a single matched strength the two updates do

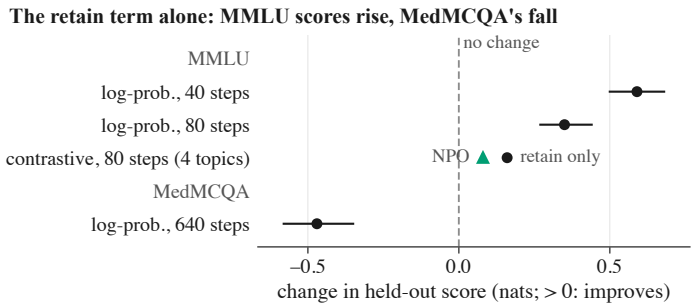


**Figure 23: The retain term alone raises held-out scores on MMLU and lowers them on MedMCQA.** Change in held-out score (after unlearning minus before; positive means held-out topics improve) for the retain-only run, with 95% intervals where available; on the contrastive score, the four topics where it ran beside NPO (triangle, NPO at 80 steps; point estimates only).

not differ measurably: the real one costs 0.000 nats more $[-0.014, 0.014]$ (11 runs on 8 targets). A random update that matches only the norm of RMU's update, and not its spectrum, does less damage than the spectrum-matched one, by 0.006 nats [0.003, 0.010] (8 targets, one strength), so matching the spectrum makes the control stricter.

RMU's held-out damage therefore grows with strength faster than that of a random update of the same size and spectrum: the directions RMU chooses, and not only the size of its update, drive how its damage scales. RMU computes these directions from the forget documents, which are identical for both copies.

A second, simpler control adds Gaussian noise to the same weight matrices that RMU updates, on the base model, at norms from 0.5 to 16 (three draws each); RMU's update at its strongest MMLU strength, (80, 190), has norm 18.4. At every norm the noise changes held-out log-probability by less than 0.001 nats and accuracy by less than 0.15 points, against 1.64 nats and 5.8 points for RMU at (80, 190). A weight change of RMU's size in a random direction does almost no held-out damage.

### F.4 Retain-only control

GradDiff and NPO combine a forget term on the forget documents with a retain term, the ordinary language-modelling loss on a small set of correct statements about other topics. Setting the forget term's weight to zero leaves the retain term running alone, with the same steps, learning rate and data: the retain-only run. RMU's retain term instead matches the frozen model's activations and has no effect on its own.

On MMLU the retain-only run improves held-out topics. On correct-answer log-probability it raises the held-out score by 0.59 nats [0.50, 0.68] at 40 steps and by 0.35 nats [0.27, 0.44] at 80 steps (21 topics). On the contrastive score, on the four topics where it was run beside an NPO run at 80 steps, it raises the held-out score by 0.16, against 0.08 for the NPO run: both move held-out scores upward, and the retain-only run moves them further. The retain-only run also never produced degenerate generation in more than 13% of its runs, at any step count up to 960 (Appendix G). On MedMCQA, at 640 steps, it instead lowers held-out log-probability by 0.47 nats [0.35, 0.58], and on eight WMDP topics by 0.37 nats [0.22, 0.51] at 640 steps and 0.28 [0.07, 0.49] at 240 steps. In accuracy, in the split-topic design on MMLU, the retain-only run raises held-out accuracy by 3.25 points [2.88, 3.62]; NPO at 80 steps is indistinguishable from it ($-0.16$ points $[-0.72, 0.40]$), and RMU at 20 steps loses 5.2 points [4.6, 5.7] more. The retain-only run also improves the held-out topics closest to $t$ the most, the reverse of RMU's pattern (Appendix D.8), at all seven step counts on Llama-3.1-8B-Instruct and at 40 steps on Qwen2.5-7B-Instruct, where it raises held-out log-probability by 0.27 nats [0.23, 0.32].

The held-out damage we measure is therefore net of the retain term, which on its own moves held-out scores (upward on MMLU), and a retain-only run is not a zero-damage baseline. Both copies receive the same objective, retain term included.

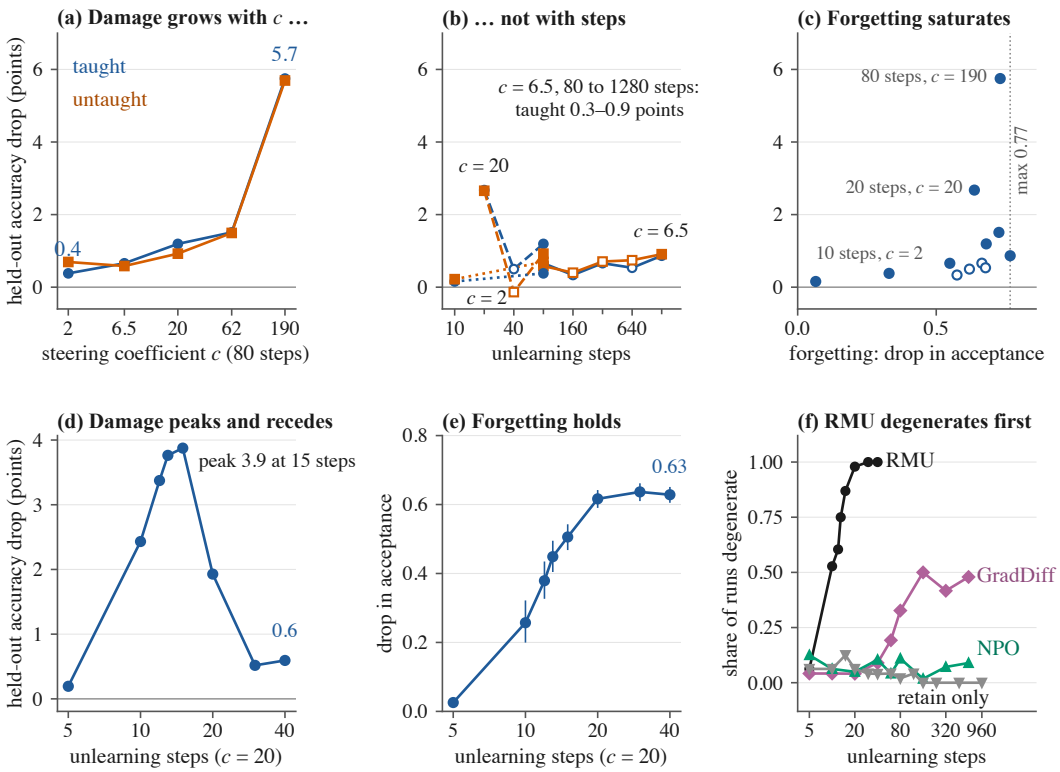


**Figure 24: Damage grows with the steering coefficient, not with the number of steps, and peaks and recedes while forgetting holds.** Llama-3.1-8B-Instruct on MMLU under RMU. Top row, the main strength grid, both copies: (a) held-out accuracy drop against the coefficient $c$ at 80 steps, for the taught (blue circles) and untaught (vermillion squares) copies; (b) the same against the number of steps at fixed $c$; hollow markers: strengths run on eight forget topics, the others on 24; (c) the taught copy's damage against forgetting, the drop in its acceptance of the taught statements, at each of the 12 strengths. Bottom row, a finer ladder of steps on taught copies at $c = 20$: (d) held-out damage peaks at 15 steps and recedes; (e) forgetting rises and levels off; (f) share of runs with degenerate generation, one fixed configuration per method. No run is screened out.

## G UNLEARNING STRENGTH

This section describes how unlearning strengths were chosen and how damage, forgetting and degenerate generation change with strength (Section 4, Figure 4).

**Strength.** For RMU, unlearning strength has two parts: the number of optimisation steps and the steering coefficient, which sets how far RMU pushes the model's activations on the forget documents. The learning rate ($5 \times 10^{-5}$), batch size and updated weights are fixed. For GradDiff and NPO, strength is the number of steps.

**How the strengths were chosen.** The strengths were chosen from earlier RMU runs on taught copies to span unlearning from barely effective to strong, and none was tuned on the damage ratio. On MMLU with Llama-3.1-8B-Instruct the grid has 12 strengths, from 10 to 1280 steps and from coefficient 2 to 190 (Figure 24); eight cover all 24 forget topics and four cover eight. MedMCQA and WMDP use four strengths each: 10 steps at coefficient 2, and 80 steps at coefficients 2, 8 and 16. The other models use three to nine strengths, with coefficients set separately for each model because the scale of the activations RMU acts on differs between models (Appendix B.5). Every strength enters the damage ratio on both copies of every topic it covers.

**Damage grows with the coefficient, not with the number of steps.** At 80 steps, raising the coefficient from 2 to 190 raises the taught copy's damage from 0.4 to 5.7 percentage points (Figure 24). At coefficient 6.5, going from 80 to 1280 steps leaves it between 0.3 and 0.9 points. At coefficient 20, it is 2.7 points at 20 steps, 0.5 at 40 and 1.2 at 80. In every row the untaught copy's damage is

close to the taught copy's. The weakest strength, 10 steps at coefficient 2, removes almost nothing (forgetting 0.07) and does almost no damage. Forgetting itself rises with both parts of strength and levels off near 0.75. This holds over the coefficients we use, 2 to 190. Below about 2 the coefficient barely changes the update: in a sweep on TOFU from $c = 0.004$ to 2 at 80 steps, the norm of the weight update changes by at most 1.1%.

**Damage rises, peaks and recedes.** A finer ladder of steps shows this shape more clearly (Figure 24). The ladder was run on taught copies only, with RMU at coefficient 20, from 5 to 40 steps on all 24 MMLU topics. Held-out damage rises from 0.2 points at 5 steps to a peak of 3.9 points at 15 steps, then falls to 0.5 points at 30 steps. Paired by topic against the 15-step run, held-out accuracy is 3.7 points higher at 5 steps [2.5, 5.1], 1.9 points higher at 20 steps [1.1, 2.9] and 3.4 points higher at 30 steps [2.2, 4.8]; these intervals resample both forget topics and held-out topics. Forgetting, the drop in acceptance of the taught statements, does not recede: it rises to about 0.6 by 20 steps and stays there. In the split-topic design (Appendix H) at $c = 2$, the taught copy's held-out log-probability damage stops growing by about 60 steps (0.078 nats at 60 steps, 0.074 at 960, with a peak of 0.089 at 120), while forgetting keeps rising: between 240 and 960 steps it rises by 0.25 [0.21, 0.29], and held-out damage on the contrastive score changes by $-0.0004$ [$-0.0010$, 0.0003]. We chose this window after seeing the runs.

**Degenerate generation grows with strength.** The run flag and topic flag of Appendix D.7 read the 16-token greedy generation for each evaluation question. Under RMU at coefficient 20 the share of flagged runs is 6% at 5 steps, 53% at 10, 87% at 15, 98% at 20 and 100% at 30 and 40 steps (Figure 24f). Under GradDiff it stays at or below 10% up to 40 steps and reaches 50% at 160 steps. NPO and the retain-only run stay at or below 13% at every step count we ran, up to 640 and 960 steps respectively. In the main experiment the topic flag fires on 1%, 6% and 11% of held-out topic scores on MMLU, MedMCQA and WMDP, far less often than the run flag.

The flags read generated text, while the damage measures score given options, so a flagged run can still be scored. Removing flagged held-out topics leaves the damage ratio close to its value on all topics (Appendix D.7). The receding part of the damage curve coincides with near-universal degenerate generation. Held-out damage is not a monotone function of strength, and the damage ratio pools a whole grid of strengths.

## H GradDiff, NPO and other methods

This section gives the full results for unlearning methods other than RMU, which Section 5.4 summarises.

**Design.** Except where marked, these results come from the split-topic design, on the same 24 MMLU topics with Llama-3.1-8B-Instruct. It differs from the main design in how the untaught copy is built. Each topic's fine-tuning documents are split into two halves, matched in length, that teach different false statements. One copy is fine-tuned on half A and the other on half B, and both are then unlearned on half A. The copy that learned half A is the taught copy. The copy that learned half B is untaught with respect to the forget documents, having learned other statements about the same topic. Each topic is run in both directions (forget A, forget B), so every half plays both roles. Damage is the fall in correct-answer log-probability on the 23 held-out topics, the damage ratio is defined as in Section 3, and the intervals are paired percentile bootstraps over topics. RMU and NPO were run in the same design. GradDiff, NPO and SimNPO update the same weights as RMU, the MLP down projections of layers 5–7, so the methods differ in their objective and not in the weights they change.

**RMU in the split-topic design.** At every RMU strength run here (5 to 960 steps, coefficients 2 and 20), the damage ratio lies between 0.93 and 1.06. The taught copy's damage is at most 0.51 nats, reached at 15 steps with coefficient 20, with a ratio of 1.01 [0.99, 1.04]. The split-topic design thus reproduces the main result for RMU. On Qwen2.5-7B-Instruct, with the same 24 topics and coefficient 20, RMU's log-probability ratio is 1.00 [0.97, 1.03] at 40 steps and 1.00 [0.98, 1.02] at 20 steps. We also ran GradDiff there, at 160 steps, but one of the 48 fine-tuned halves missed the

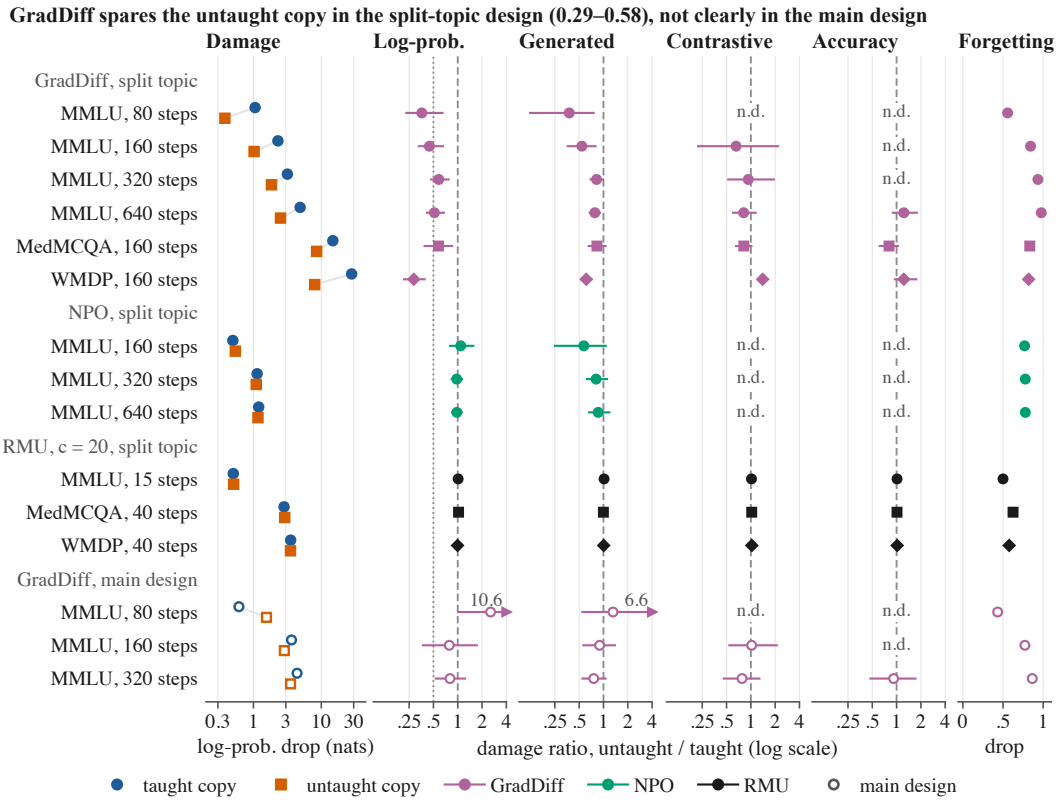


**Figure 25: Under GradDiff the untaught copy takes 0.3 to 0.6 of the taught copy's log-probability damage on all three datasets in the split-topic design, and not clearly less in the main design; under NPO, at the same amount of damage, and under RMU, it takes as much.** Llama-3.1-8B-Instruct; split-topic design (filled) and, for GradDiff on MMLU, the main design (hollow). Damage: fall in held-out correct-answer log-probability of the taught (blue circles) and untaught (vermillion squares) copies (log scale). Next four panels: damage ratio (log scale) with its 95% paired bootstrap interval over topics on four measures, for GradDiff (purple), NPO (green) and RMU (black); markers give the dataset (circle MMLU, square MedMCQA, diamond WMDP); the dotted line marks one half. Interval ends beyond the axis are arrowheads with their value printed. *n.d.*: not defined, because the taught copy's damage has an interval that includes zero or lies below it. Forgetting: drop in the taught copy's acceptance of the taught statements. GradDiff and NPO rows on MMLU are the strengths with log-probability damage above zero, the RMU row is RMU's most damaging MMLU strength, MedMCQA and WMDP have one strength per method, and the main-design rows are the three strengths run there.

learning check fixed before the runs (acceptance 0.48 against a threshold of 0.50), so we set that comparison aside, and GradDiff's split-topic result is from Llama-3.1-8B-Instruct.

**GradDiff.** Figure 25 gives GradDiff at the four strengths where the taught copy's damage is above zero: the damage ratio is 0.36, 0.45, 0.58 and 0.51. Every upper limit is below one, so the untaught copy is damaged less than the taught copy, and every interval contains 0.55, so one ratio of about one half fits all four strengths. The generated-answer accuracy ratios at the four strengths are 0.38, 0.54, 0.82 and 0.78, all with intervals below one; the contrastive-score ratios at the three strongest are 0.66, 0.93 and 0.81, with intervals that contain one. Restricted to questions that both copies answered correctly before unlearning, the generated-answer accuracy ratios are 0.65, 0.68, 0.86 and 0.81. At weaker GradDiff strengths (5 to 40 steps) the taught copy's held-out log-probability rises, so damage is negative and the ratio is not defined.

**MedMCQA and WMDP.** We also ran the split-topic design on the 24 MedMCQA and 24 WMDP topics, with one strength per method set from calibration runs before these runs: GradDiff at 160 steps and RMU at 40 steps with coefficient 20, both with the retain set of the main study. GradDiff again spares the untaught copy: its log-probability ratio is 0.58 [0.39, 0.85] on MedMCQA and 0.29

[0.22, 0.39] on WMDP, and its generated-answer accuracy ratio 0.83 [0.66, 1.06] and 0.61 [0.53, 0.71]. RMU's log-probability ratio is 1.02 [1.00, 1.05] and 1.00 [0.98, 1.01], and its generated-answer accuracy ratio 1.00 and 1.01. On WMDP the difference between the two methods' log-probability ratios, fixed as the test before any of these runs, is 0.71 [0.60, 0.78]. As on MMLU, GradDiff does far more damage at these strengths: its taught copy loses 14.8 and 28.0 nats of held-out log-probability, against 2.8 and 3.5 for RMU. On the contrastive score, GradDiff's ratio is 0.82 [0.66, 1.02] on MedMCQA and 1.40 [1.23, 1.60] on WMDP, and RMU's is 1.03 on both.

A second WMDP block, on 16 topics not used to set any strength, ran RMU at a weaker setting, 240 steps with coefficient 4, beside GradDiff at 160 steps. It gives the same separation: RMU's log-probability ratio is 1.01 [1.00, 1.03] and GradDiff's 0.24 [0.17, 0.33], a difference of 0.78 [0.69, 0.84]; in generated-answer accuracy they are 0.94 [0.88, 1.01] and 0.55 [0.47, 0.63], and on the contrastive score 1.07 [1.02, 1.13] and 1.18 [1.00, 1.40]. The taught copy loses 1.01 nats under RMU and 34.2 under GradDiff. RMU's setting was chosen by a rule that required fewer than half of the calibration runs to show degenerate generation, a threshold on which three candidate settings sat exactly; in this block 49 of RMU's 64 runs show degenerate generation, and 22 of GradDiff's.

The calibration blocks for the MedMCQA and WMDP runs, on eight MedMCQA and eight WMDP topics, compare two forget sets on one model: a copy that learned half A of a topic is unlearned either on half A or on half B, which it never learned. GradDiff at learning rate $5times10^{-5}$ spares the copy unlearned on the half it never learned at every strength from 160 to 1280 steps, with log-probability ratios from 0.13 to 0.66 and intervals below one at six of the eight dataset–strength combinations. RMU at coefficient 20, from 40 to 320 steps, gives 1.00 to 1.11. At learning rate $2times10^{-4}$, where GradDiff's taught-copy damage reaches 64 to 121 nats and every run shows degenerate generation, its ratio is 0.79 to 1.28.

**GradDiff in the main design.** We also ran GradDiff in the main design, where the untaught copy learned the unrelated topic $\sigma(t)$ (24 MMLU topics, 80, 160 and 320 steps, with the same settings as in the split-topic runs; hollow markers in Figure 25). There GradDiff spares the untaught copy much less. In log-probability the ratio is 2.55 [1.02, 10.6] at 80 steps, where the taught copy loses 0.61 nats and the untaught copy 1.56, and 0.79 [0.37, 1.72] and 0.80 [0.54, 1.23] at 160 and 320 steps, where the taught copy loses 3.6 and 4.4 nats. Generated-answer accuracy gives 1.31 [0.55, 6.6], 0.90 [0.56, 1.39] and 0.76 [0.55, 1.06]. In accuracy the taught copy gains 1.8 points at 80 steps, the ratio is not defined at 160 steps, and it is 0.92 [0.47, 1.71] at 320 steps, where the taught copy loses 2.3 points. The taught statements are removed as in the split-topic runs: the taught copy's acceptance falls by 0.43, 0.77 and 0.86. GradDiff's sparing of the untaught copy therefore appears when that copy has learned other statements about the forget topic, and not clearly when it has learned an unrelated one.

**Accuracy in the split-topic design.** In accuracy the split-topic runs do not separate the methods. At most strengths GradDiff and NPO raise the taught copy's held-out accuracy, through their retain term (Appendix F.4), so the damage ratio is not defined. Where GradDiff lowers it, the accuracy ratio is 1.23 [0.90, 1.78] on MMLU at 640 steps, 0.81 [0.62, 1.04] on MedMCQA and 1.23 [0.95, 1.75] on WMDP, all with intervals that contain one, and RMU's accuracy ratios lie between 0.71 and 1.22. GradDiff's sparing of the untaught copy is therefore a result in log-probability and generated-answer accuracy.

**Matching the amount of damage.** GradDiff's smallest taught-copy damage among these four strengths, 1.06 nats, is twice RMU's largest, 0.51, so the two methods differ both in how they unlearn and in how much damage they do. To separate the two, we ran GradDiff at a strength chosen in a pilot run to match RMU's damage. Its taught-copy damage is 0.30 nats [$-0.06$, 0.70], against 0.29 [0.23, 0.35] for RMU at 20 steps. We compare the methods on the difference between the taught and untaught copies' damage. If the method mattered and the amount of damage did not, GradDiff's difference should exceed RMU's by 0.13 to 0.19 nats. The measured gap is $-0.02$ [$-0.44$, 0.39], an interval that contains both zero and the predicted range.

Generated-answer accuracy separates the two methods more clearly at this damage level. At the damage-matched strength (60 steps), GradDiff's taught copy loses 0.6 points and its untaught copy 0.1, a difference of 0.6 [$-0.1$, 1.2]; in a re-run on the 21 topics not used for calibration, 1.0 against 0.1 points, a difference of 0.8 [0.1, 1.7]. RMU at the same log-probability damage (20 steps) gives a difference of 0.05 [$-0.07$, 0.18] points.

The test is limited by the spread across topics. At this strength GradDiff's taught-minus-untaught difference has a standard deviation of about 1.1 nats across topics, and detecting a gap of the predicted size with 80% power would need 235 to 554 topics. Only 44 topics have enough documents to build the two halves. Between 33% and 80% of the variance of the difference lies between topics, so repeating runs helps little. Nor can the comparison move to a higher damage level: RMU's held-out damage peaks below the level at which GradDiff's ratio first becomes defined (Appendix G).

**NPO.** At 10 to 80 steps, in each of the six configurations we ran, the taught copy's held-out log-probability damage is negative or has an interval that contains zero, so the damage ratio is not defined. Generated-answer accuracy behaves differently: at 40, 60 and 80 steps the taught copy loses 0.7 to 1.3 points and the untaught copy none, a paired difference with an interval above zero in all four such configurations. At these steps the two copies are not unlearned equally deeply: at 40 steps the loss on the forget documents rises by 4.3 nats per token in the taught copy and by 13.1 in the untaught copy. A second block of runs at 160, 320 and 640 steps reaches taught-copy damage of 0.50, 1.13 and 1.20 nats, with forgetting of 0.77 to 0.78 (Figure 25). The damage ratio is 1.09 [0.80, 1.56], 0.97 [0.84, 1.13] and 0.98 [0.85, 1.12], and the paired difference is $-0.04$ [$-0.19$, 0.13], 0.03 [$-0.13$, 0.21] and 0.03 [$-0.12$, 0.20] nats. The generated-answer accuracy ratios are 0.57, 0.81 and 0.86, with intervals that contain one. On the contrastive score, held-out scores rise in both copies at all three strengths, by about 0.08 nats more in the taught copy, so the ratio is not defined. About 4% of runs show degenerate generation at each strength. At 320 steps NPO does as much damage to the taught copy as GradDiff at 80 steps (1.13 against 1.06 nats), and its ratio, 0.97 [0.84, 1.13], lies well above GradDiff's, 0.36 [0.23, 0.65]: at the same amount of damage, GradDiff's damage depends on what the model learned and NPO's does not. These two settings were matched after the runs; NPO's ratio is close to one at all three of its strengths, so the conclusion about NPO does not rest on the match.

On eight WMDP topics we also ran NPO (240 steps) in the calibration design above, which compares two forget sets on one model. There, unlearning the half the model learned costs it more held-out log-probability than unlearning the half it never learned: the ratio is 0.69, with a paired difference of 1.18 nats [0.01, 2.52], and in generated-answer accuracy 0.67 [0.49, 0.97]; on the contrastive score it is 1.01 [0.66, 1.75]. RMU in the same block gives log-probability ratios of 1.03 to 1.07.

**SimNPO.** At 10, 20 and 40 steps SimNPO did not remove the taught statements: the drop in acceptance is between $-0.002$ and 0.011, and held-out scores rise by the same amount in both copies. A separate block at 80 steps on the 24 MMLU topics, run beside RMU at the same steps (coefficient 20), does remove them: the taught copy's acceptance falls by 0.57. There the two copies part ways. The taught copy loses 0.14 nats of held-out log-probability [$-0.02$, 0.30], while the untaught copy gains 0.14 [0.05, 0.23], a paired difference of 0.27 nats [0.12, 0.44]; in generated-answer accuracy the difference is 0.8 points [0.5, 1.2]. The ratio is not defined, because the taught copy's damage interval contains zero, and on the contrastive score both copies improve by about the same amount (difference 0.01 [$-0.01$, 0.04]). RMU in the same block gives a ratio of 1.00 [0.91, 1.10] and a paired difference of 0.00 [$-0.01$, 0.01] nats, but every one of RMU's runs in this block shows degenerate generation. The 10- to 40-step SimNPO runs used the published setting ($\beta = 4.5$, forget weight 0.125, retain weight 1), which ranked ninth of the ten settings we swept, at 40 and 80 steps, in how much of the taught statements it removed; the 80-step block used the setting that removed the most ($\beta = 0.3$, forget and retain weights 1). Like GradDiff, SimNPO damages the copy that learned the forget documents more than the one that learned other statements about the topic.

**Summary.** Under RMU the damage ratio is close to one in both designs, and under NPO it is close to one on MMLU wherever its log-probability damage is positive; under GradDiff it is about one half in the split-topic design and 0.79 to 0.80, with intervals that contain one, in the main design; SimNPO, at the one strength where it forgets, damages the taught copy more than the untaught one. The ratio has to be measured for each method and design.

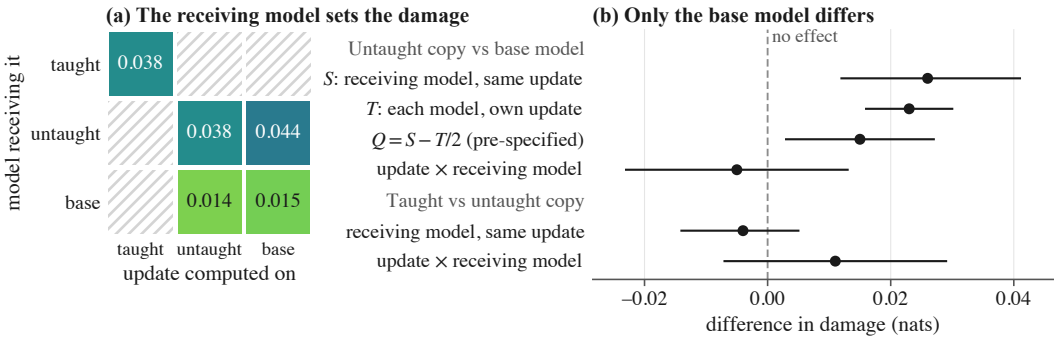


**Figure 26: The receiving model matters more than the update, but not between the taught and untaught copies.** RMU updates (1280 steps, $c = 6.5$) recorded on Llama-3.1-8B-Instruct for the 24 MMLU forget topics and applied to other models. (a) Held-out damage (contrastive score, nats) when the update computed on the column's model is applied to the row's model; hatched combinations were not run. (b) Contrasts with 95% jackknife intervals over the eight domains: from the $2 \times 2$ table of panel (a) on the untaught copy and the base model, and from a second $2 \times 2$ table on the taught and untaught copies (360 runs).

# I TRANSPLANTING WEIGHT UPDATES

This section asks whether held-out damage depends on the weight update that unlearning produces or on the model that receives it, and gives the bound on the receiving model cited in Section 5.1.

**Design.** An unlearning method computes its update from the model it is applied to, so two models that learned different things also receive different updates. We record the weight update of each run and apply it to other models, using RMU (1280 steps, coefficient 6.5) on Llama-3.1-8B-Instruct and the 24 MMLU forget topics. The untaught copy and the base model each receive both their own update and the other's, which gives a $2 \times 2$ table in which update and receiving model vary independently. The taught copy receives only its own update. Damage is the fall in the held-out contrastive score (Appendix C), averaged over topics, over 472 runs; intervals are 95% jackknife intervals over the eight domains.

**Results.** The receiving model matters more than the update (Figure 26). Holding the update fixed and averaging over the two updates, the untaught copy is damaged more than the base model by $S = 0.026$ [0.012, 0.041], and the difference is positive in all eight domains (sign test $p = 0.008$). Re-running the whole experiment at a different numerical precision changes $S$ by 0.001. When each model receives its own update, the difference is $T = 0.023$ [0.016, 0.030]. The pre-specified quantity, the excess of $S$ over half of $T$, is $Q = S - T/2 = 0.015$ [0.003, 0.027]. It is positive: the receiving model accounts for more than half of the difference between the two models' own runs, and $S$ is about as large as $T$. The interaction between update and receiving model is $-0.005$ [$-0.023$, 0.013] and is not resolved.

**The main comparison.** The contrast above is between the untaught copy and the base model, whereas the main result compares the taught and untaught copies. A second $2 \times 2$ table on that pair, with 360 runs on the same 24 topics, puts the effect of the receiving model at $-0.004$ [$-0.014$, 0.005]; the design has power 0.24 to detect an effect of $\pm 0.010$. The interaction is 0.011 [$-0.007$, 0.029]. Between the taught and the untaught copy, the effect of the receiving model, averaged over the two updates, is bounded below about 0.014.

**Reading.** Being fine-tuned at all makes a model more fragile to the same update: both copies were fine-tuned for 200 steps on up to 50 documents and the base model was not. Between the taught and untaught copies, which differ in the content they learned, the experiment finds no difference, averaged over topics.

## J EXTENDED RELATED WORK

This section places the untaught-copy control among closely related work in more detail than Section 2. Surveys of LLM unlearning (Liu et al., 2025; Barez et al., 2025) review the main families of methods, from gradient ascent (Jang et al., 2023) and gradient ascent with a retain term (Yao et al., 2024; Liu et al., 2022) to representation steering (Li et al., 2024) and preference-based losses (Zhang et al., 2024; Fan et al., 2025a).

**How collateral damage is reported.** Unlearning benchmarks pair a score for forgetting with a score for what is kept. TOFU (Maini et al., 2024), MUSE (Shi et al., 2025), WMDP (Li et al., 2024) and RWKU (Jin et al., 2024) all measure the second score on held-out data, comparing the model after unlearning with the same model before it or with a model trained without the forget data. OpenUnlearning collects many methods and metrics under one harness (Dorna et al., 2025), and we run our unlearning through it. Recent work studies where the damage falls. PreUnlearn measures collateral damage against the model before unlearning and finds it strongest near the forget set, weakening with semantic distance without vanishing (Su et al., 2026). RippleBench measures ripple effects on knowledge related to the forget target (Rinberg et al., 2025), and knowledge editing has a similar line of work, in which edits spread to related facts (Cohen et al., 2024) and degrade general abilities as they accumulate (Gu et al., 2024; Gupta et al., 2024). DUSK and BLUR build forget and retain sets that share content and find that current methods either erase shared knowledge or fail to forget (Jeung et al., 2026; Hu et al., 2025b); KnowUnDo reports that existing methods remove knowledge that should be kept (Tian et al., 2024), and in classifiers, a recent method targets damage on retained data entangled with the forget set (Cheng et al., 2026). Current methods asked to remove a fact together with the facts from which it can be deduced either fail to remove it or remove many unrelated facts (Wu et al., 2024). MNEME predicts side effects of WMDP unlearning, such as lost chemistry accuracy after biology is removed, from the difference between the models before and after (Kassem et al., 2025), and PrivUn finds that the ripple effects of privacy unlearning follow gradient-based associations rather than semantic ones (Chen et al., 2026a). Wybitul et al. (2026) vary how entangled a model's representations are while holding the data and the unlearning method fixed, and find that entanglement raises the cost on retained knowledge; our control instead varies whether the target was learned.

Unlearning also leaves traces that can be detected on inputs unrelated to the target (Chen et al., 2026b). RMU pushes the representations of forget inputs toward a random vector, which lowers the model's confidence on them (Dang et al., 2025). RMU- and NPO-style unlearning make models misbehave on retain queries that contain a forget token (Dang et al., 2026a), and the side effects of representation misdirection are set by its target vector (Dang et al., 2026b), which fits our finding that RMU's damage does not require the target to have been learned. Outside language models, Dolgova & Stich (2026) compare the unlearned model with a retrained one example by example and find damage concentrated near the forget set. All of these read damage relative to the model before unlearning or to a retrained model. Our control asks how much of that damage the same unlearning run causes in a model that never learned the target.

**Critiques of unlearning evaluation.** Several papers argue that unlearning benchmarks measure less than they claim. Thaker et al. (2025) argue that current benchmarks are weak measures of progress. Feng et al. (2025) show that existing evaluations can both overstate and understate unlearning. Cooper et al. (2025) separate what unlearning methods do from what policy expects of them. Lynch et al. (2024) list evaluations that robust unlearning should pass, including side effects on similar domains, and Doshi & Stickland (2024) find that LLMU and RMU lower general capabilities, that few-shot prompting or simple rephrasing can raise accuracy on the forgotten questions more than tenfold, and that fine-tuning on unrelated data almost fully restores it. Fine-tuning-based unlearning changes how knowledge is retrieved rather than erasing it, and the change reaches unrelated knowledge (Hong et al., 2024). Fan et al. (2025b) show that multiple-choice accuracy on WMDP gives a narrow view of unlearning and group methods into divergence-driven, representation-misalignment and rejection-based families. Work on relearning and adversarial recovery shows that unlearned knowledge often returns after small amounts of fine-tuning, quantisation or targeted prompting (Hu et al., 2025a; Łucki et al., 2025; Zhang et al., 2025; Che et al., 2025), which tamper-resistant training aims to prevent (Tamirisa et al., 2025). These critiques concern forgetting; ours concerns what a damage number measures.

**Models that never saw the forget data.** TOFU trains a retain model that never sees the forget set (Maini et al., 2024). It is the target that unlearning should reach and is compared with the unlearned model on forgetting, but it is not itself unlearned. OpenUnlearning's meta-evaluation of metrics also uses models trained with and without the forget material, to test whether a forgetting metric can tell them apart (Dorna et al., 2025). O'Brien et al. (2026) filter pretraining data to build models that never saw hazardous knowledge and compare them with unlearning after training, and Yang & Yeung (2026) compare unlearned models with a matched model that never learned the forget facts. There the question is whether knowledge is present; ours is how much held-out damage requires the knowledge to have been learned.

**Unlearning where there is nothing to remove.** The closest work is Kim et al. (2026). They mix question–answer pairs that a TOFU-fine-tuned model never saw into its forget set and show that, under gradient ascent, their gradient analysis shows these unseen items dominating the damage on the retain set. They compare items within one model and conclude that forget sets should be aligned with what the model knows. We instead compare two copies under the same forget set and read the damage number against the untaught copy's damage: under RMU, and under NPO at GradDiff's damage level, the two copies are damaged equally, and under GradDiff the untaught copy is damaged less when it learned other documents about the forget topic. Their argument concerns the gradients of gradient ascent, whereas RMU's loss acts on representations. A gradient analysis of unlearning objectives, RMU among them, traces how each objective moves performance on data other than the target (Wang et al., 2025b); it does not compare against a model that never learned the target. In image classifiers, adversarial requests to unlearn data the model was never trained on can destroy its accuracy (Huang et al., 2025).

In vision–language models, SALMUBench trains a clean encoder that never saw a sensitive set and applies several unlearning baselines to it, motivated by the tendency of unlearning to damage a model whether or not the sensitive data were present (Selvas-Sala et al., 2026). Its check reports how that encoder moves. Zhao et al. (2025) apply their backdoor-removal procedure to models without a backdoor to check that it does not hurt clean accuracy. Sadhu et al. (2026) run their unlearning on randomly chosen layers as a control, which holds the target fixed and moves where the update lands. Our control holds the update procedure fixed and moves what was learned. Our contribution is a paired construction on one base model, with fine-tuning volume and the unlearning run held fixed, that turns the idea into an estimate of how much of a reported damage number requires the target to have been learned. In image generation, Saha et al. (2025) study side effects of concept erasure on neighbouring concepts.

**Knowledge introduced by fine-tuning.** Our taught copy learns the forget topic through fine-tuning, as in TOFU and similar benchmarks. Deeb & Roger (2024) find that unlearning evaluations built on knowledge added by fine-tuning may overestimate how robustly unlearning removes it, compared with knowledge learned in pretraining. Their finding concerns forgetting; for collateral damage, unlearning what the base model knows from pretraining costs as much as unlearning statements it never saw (ratio 1.04; Appendix E). Ye et al. (2026) show that fine-tuning on counterfactual statements, used as an unlearning method, spreads fabrication to unrelated domains. We use false statements only to teach, and both copies are fine-tuned on them for the same number of steps. Teaching new facts by fine-tuning has side effects of its own, such as more hallucination (Gekhman et al., 2024), and LoRA fine-tuning forgets less of what the base model knew than full fine-tuning (Biderman et al., 2024); both copies receive the same LoRA fine-tuning, so such effects are shared. Pretrained and fine-tuned models also respond differently to unlearning: when the forget facts were first learned by fine-tuning, forgetting is smoother and more of the remaining knowledge is kept (Borisiuk et al., 2026), and the effect of unlearning depends on the order in which a model was trained (Yu et al., 2025), one reason both copies follow the same recipe.

**The retain term.** Most unlearning objectives include a retain term, and how the retain set is chosen affects what is kept (Chang & Lee, 2025). Wang et al. (2025a) calibrate methods to equal retention before comparing forgetting, which still reads retention against the original model. Retain-only fine-tuning can raise held-out scores for reasons unrelated to forgetting (Feng et al., 2025), and some work does not count movement above the original model, reading it as a benign side effect or noise (Yüce et al., 2026). Siddiqui et al. (2025) find, in image classifiers, that fine-tuning on the

retain set alone can restore forget-set accuracy. Our own retain-only runs raise held-out scores on MMLU (Appendix F.4).

**Controls that remove the cause.** Other fields answer the same question with the same kind of comparison. Surgical trials compare real surgery with a sham procedure that repeats everything except the step thought to help (Moseley et al., 2002). Epidemiology uses exposures that cannot cause the outcome to detect bias (Lipsitch et al., 2010). Saliency methods are tested on models whose weights have been randomised; a method whose maps do not change is insensitive to what the model learned (Adebayo et al., 2018). Concept erasure checks that an intervention's side effects are specific to what it removes, for example by comparing the removal of a property with the removal of random directions of the same size (Elazar et al., 2021). Probing results are read against control tasks that assign random labels to word types, so that accuracy on them reflects the probe rather than the representation (Hewitt & Liang, 2019), and the ability of networks to fit random labels changed how their generalisation is explained (Zhang et al., 2017). The untaught copy plays the same role for collateral damage: it keeps the whole unlearning procedure and removes only the learned target.

## K Reproducibility

This section lists what is needed to reproduce the paper: the software, the compute, the training seeds and the data we will release.

**Software.** Unlearning runs through OpenUnlearning (Dorna et al., 2025), pinned to a fixed upstream version, and every method is run through the same training loop. The two copies are fine-tuned with LoRA (rank 16, $\alpha = 32$) for 200 steps. Each run stores its full configuration and its input data with its results.

**Compute.** All runs used NVIDIA L40S GPUs (48 GB), one GPU per run. A typical run on Llama-3.1-8B-Instruct, which fine-tunes one copy, unlearns it and scores it, takes about 5.5 minutes, of which about 70% is scoring. The project stored about 30,000 scored runs, including exploratory and superseded ones; at the per-run time above, the total is on the order of 3,000 GPU-hours. The main MMLU experiment for Llama-3.1-8B-Instruct is 224 (topic, strength) runs on each copy.

**Training seeds.** The main runs use fine-tuning seed 0, and each (copy, topic, strength) combination is unlearned once, except on the main MMLU grid, where all 224 runs of both copies were executed a second time with identical settings (for the taught copies, as part of the second pairing of Appendix B.3). The repeated runs are not bit-for-bit identical to the first, and they diverge more often at more steps: of the same eight untaught runs at coefficient 6.5, 4 differ from their first execution at 80 steps and all 8 at 1280 steps. The second execution gives the same result as the first (Appendix E). For Llama-3.1-8B-Instruct on MMLU we repeated the experiment with fine-tuning seeds 1 to 13 (Appendix D.9). The random choices of the positive control re-draw which topics are targeted while the training recipe stays fixed (Appendix F.1).

**From records to numbers.** Figure and table values are computed from the stored result records, and numbers in the text were checked against them.

**Release.** We will release the fine-tuning and forget documents, the topic lists and pairings, and the stored result records after the review period. The documents were generated for this study; those with false statements about benchmark topics are marked as such in the release.