DETER: SUPPRESSION DURING PRETRAINING FOR TAMPER RESISTANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Open-weight language models can acquire knowledge that could enable biological, chemical, or cyber misuse. Any attempt to limit such knowledge must remain effective after the model weights are released and can be fine-tuned. Unlearning applied after pretraining can be rapidly undone, whereas filtering the pretraining corpus offers greater resistance. We ask whether actively suppressing knowledge during pretraining can further increase the fine-tuning required to acquire it. Even simple gradient ascent interleaved with standard pretraining makes medical knowledge harder to acquire through adversarial fine-tuning than filtering alone or post-hoc unlearning. At 0.5B parameters, reaching the held-out PubMed loss of an unfiltered model requires over 13$\times$ as many PubMed fine-tuning tokens as after filtering alone. Retained knowledge is evaluated using both held-out data and downstream tasks. Across model sizes, varying suppression strength reveals an increasingly favorable trade-off between fine-tuning difficulty and held-out loss on retained pretraining data. These results point to pretraining as an effective stage for making unwanted capabilities harder to acquire through subsequent fine-tuning.

1 INTRODUCTION

Open-weight models pose a particular challenge for machine unlearning: after a model has been modified to suppress knowledge from a target (or *forget*) domain, an adversary with access to the weights can fine-tune it directly on data from that domain (Henderson et al., 2023). *Tamper resistance* measures how well the effect of unlearning persists under such adversarial fine-tuning, or equivalently, how difficult it is to recover the suppressed target-domain capability (Tamirisa et al., 2025). Existing post-training unlearning methods are often brittle under this threat model, with target-domain performance recovering after relatively little additional fine-tuning (Qi et al., 2025; Lee et al., 2025; Che et al., 2025). Pretraining data filtering offers a substantially more durable alternative (O'Brien et al., 2026). For knowledge-intensive domains, filtering limits capability acquisition by removing content associated with the target domain before training. The filtering rule must balance reducing target-domain learning against preserving performance on the non-target portion of the training distribution, the *retain distribution*. Finer-grained removal at the token level improves this trade-off (Rathi & Radford, 2026), and stronger supervision may come from measuring each document's influence (Yu et al., 2024) on the target domain relative to the retain distribution. Yet these refinements still estimate which data contribute to target-domain learning, and remain a proxy for what knowledge the final model will ultimately acquire from the retained corpus over the course of pretraining. Target-domain learning can therefore persist even under sophisticated data selection.

Filtering provides no explicit signal about how the model should behave on the target domain; its behavior there emerges from training on the retained corpus. The highest-scoring documents removed by filtering can provide an explicit negative training signal, forming a second layer of defense against target-domain learning (Figure 1). We ask whether actively opposing target-domain learning during pretraining makes a model more resistant to subsequent target-domain fine-tuning than document filtering alone or post-hoc unlearning.

To investigate this question, we pretrain language models of up to 0.5B parameters from scratch, using medical knowledge as a proxy target domain following Rathi & Radford (2026). Medicine provides a knowledge-intensive domain with measurable capability at these model scales and a closely

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.

related retain domain, biology, against which to evaluate the selectivity of suppression. We filter medical documents out of FineWeb-Edu and interleave ordinary pretraining on the retained corpus with suppression updates on a subset of the removed documents. We count how many PubMed fine-tuning tokens it takes to bring each model back to the PubMed loss of the unfiltered baseline. Suppression during pretraining more than doubles this number compared with both filtering alone and Representation Misdirection for Unlearning (RMU; Li et al., 2024), a post-hoc unlearning method that steers target-domain representations toward a random vector while preserving representations on retained data (Section 3).

Crucially, the observed gains are achieved with simple gradient ascent, despite its documented tendency to degrade general capabilities in post-hoc unlearning (Zhang et al., 2024). When interleaved with pretraining, this simple objective adds substantial fine-tuning difficulty with limited degradation in retain performance, as measured by corpus and task loss. Just as pretraining lays the foundations for later learning, we argue that it should also lay the foundations for resistance to unwanted learning. Understanding how to build capabilities and how to restrict their subsequent acquisition should therefore be studied together, rather than treating suppression solely as a correction to a finished model.

Contributions. Our primary contributions are:

- We conduct a controlled empirical study of suppressing targeted knowledge during pretraining, comparing against data filtering and post-hoc unlearning across model scales and intervention choices.
- We find that suppression during pretraining makes targeted knowledge substantially harder to acquire through adversarial fine-tuning than filtering or post-hoc unlearning, with limited degradation of retained capabilities.
- We find favorable scaling behavior in the trade-off between target-domain fine-tuning difficulty and general loss: as model size increases, the suppression frontier shifts toward requiring more fine-tuning at the same general-loss cost.

2 METHODOLOGY

In this section, we describe the setting and how suppression is interleaved with pretraining. Appendix B.2 gives the exact training procedure and implementation details.

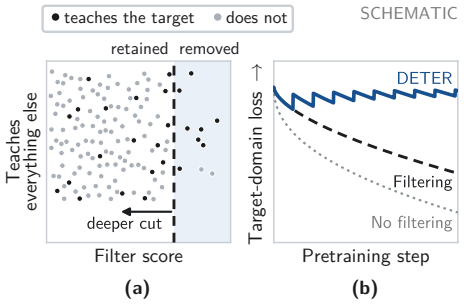


Figure 1: **Conceptual overview.** (a) Filtering decides before pretraining: its selection still contains data that teach the *target domain* along with *retain capabilities*, and a deeper cut removes more of both. (b) Adding periodic suppression provides a second layer of defense to prevent learning of the *target domain*.

Setup. We use *target capability* to refer to a capability we seek to suppress whose acquisition requires exposure to data from a *target domain*, and *retained capabilities* to those we aim to preserve. Let $\mathcal{D}$ denote the pretraining corpus and $\mathcal{D}_F \subseteq \mathcal{D}$ a set of target-domain documents. Let $\mathcal{D}_\rho \subseteq \mathcal{D}$ denote the filtered corpus obtained by removing the documents that a scorer for target-domain content ranks highest, until they account for a fraction ρ of the tokens in $\mathcal{D}$. $\mathcal{D}_F$ is a subset of the removed documents.

Let θ_t denote the parameters after t pretraining steps, with final parameters θ_T, and let $\bar{\theta}_T$ denote a reference model obtained by standard pretraining on the unfiltered corpus $\mathcal{D}$ for the same number of steps. To measure resistance, an adversary fine-tunes θ_T on target-domain data. Following Rosati et al. (2024), we measure the fine-tuning compute required to reach the target-domain loss of $\bar{\theta}_T$. We evaluate each model both on resistance and on how well it preserves retained capabilities.

Suppression during pretraining. Standard pretraining minimizes the next-token prediction loss ℓ_{pre} on $\mathcal{D}_\rho$ with learning rate η_t. Starting at step t_{start}, we take S suppression steps on $\mathcal{D}_F$ every P pretraining steps (Figure 8). For a suppression objective $\mathcal{L}_F$, the suppression steps taken at step t apply

$$\theta^{(s+1)} = \text{Update}\Big(\theta^{(s)}, \nabla_\theta \mathcal{L}_F(\theta^{(s)}); c\eta_t\Big), \qquad \theta^{(0)} = \theta_t, \quad s = 0, \ldots, S-1, \tag{1}$$

where Update denotes one optimizer step, c scales the suppression learning rate, and pretraining continues from $\theta^{(S)}$. Suppression is thus specified by the start step t_{start}, the interval P, the number of suppression steps S, the learning-rate multiplier c, and the objective $\mathcal{L}_F$. Unless stated otherwise, $\mathcal{L}_F$ is the negative next-token loss on $\mathcal{D}_F$, that is, gradient ascent (Jang et al., 2023); Section 4 considers other objectives.

Optimizer States. Pretraining and suppression use the same optimizer types for the same parameter groups and update the same model parameters but maintain separate optimizer states throughout training. For `AdamW`, pretraining and suppression each maintain their own first- and second-moment estimates. For `Muon`, they each maintain their own momentum buffer for the attention and MLP weight matrices, together with separate `AdamW` moment estimates for the parameter groups optimized with `AdamW`. These quantities are updated only by gradients from the corresponding objective and persist throughout training. Thus, the two objectives interact through the shared model parameters only.

3 EXPERIMENTS

To evaluate the effect of suppression during pretraining, we train decoder-only transformers with 113M, 224M and 521M parameters from scratch, with medical knowledge as the target domain. The architecture and hyperparameters are detailed in Appendix B.1.

3.1 SETUP

Setup. Following Rathi & Radford (2026), we pretrain on FineWeb-Edu (Penedo et al., 2024). A document classifier ranks its documents by medical content, and we remove the highest-scoring documents up to $\rho = 20\%$ of corpus tokens. $\mathcal{D}_F$ is the set of documents in the top 5% of corpus tokens. Suppression runs from the start of pretraining ($t_{\text{start}} = 0$): every $P = 50$ pretraining steps, we take a suppression step on $\mathcal{D}_F$ ($S = 1$). We pretrain and suppress with `Muon` (Jordan et al., 2024), using `AdamW` (Loshchilov & Hutter, 2019) for the embedding and normalization parameters; μP transfers one tuned peak learning rate across the three widths (Yang et al., 2021). We train each model with approximately 20 pretraining tokens per parameter, following the compute-optimal scaling of Hoffmann et al. (2022). Experiments were conducted using 1× H100 80GB or B300 288GB GPU at 113M, and 2× H100 80GB or B300 288GB GPUs at 224M and 521M.

Baselines. We compare against three baselines: (1) FILTERING, pretraining on $\mathcal{D}_\rho$ at $\rho = 20\%$ without suppression; (2) RMU (Li et al., 2024), the post-hoc method most commonly used as a baseline in evaluations of tamper resistance (Tamirisa et al., 2025; Che et al., 2025; Lee et al., 2025; Shilov et al., 2025; Rathi & Radford, 2026), applied to FILTERING after pretraining with the same number of target-domain tokens that suppression uses during pretraining; and (3) TOKEN FILTERING (Rathi & Radford, 2026), which filters medical content at the token level during pretraining, without suppression. We use its removal variant, which replaces medical tokens with a <|hidden|> token and masks the loss on them. We use a 15% filtering rate, selected at 113M to match the general loss of suppression and held fixed across model sizes. We report all losses relative to NO FILTERING, i.e., standard pretraining on the full corpus $\mathcal{D}$. Implementation details are provided in Appendix D.1.

3.2 EVALUATION

Corpus loss. Following Rathi & Radford (2026), we use held-out cross-entropy loss as a proxy for capability on target and retained domains. We measure medical and general loss on held-out FineWeb-Edu documents labeled by the document classifier: medical (documents scoring at or above the threshold that selects $\mathcal{D}_F$) and general (documents that survive filtering at $\rho = 20\%$, i.e. documents in $\mathcal{D}_\rho$). We further report medical loss on PubMed (Kandpal et al., 2025), biology loss on bioRxiv abstracts as the retained domain closest to medicine, and loss on arXiv abstracts (Kandpal et al., 2025) and PhilPapers (Gao et al., 2020) to measure broader retained knowledge, following Rathi & Radford (2026) (Appendix D.2).

Task Loss. We complement corpus loss with downstream evaluations chosen to cover the target domain, the closest retained domain, and broader retained knowledge. We evaluate medical

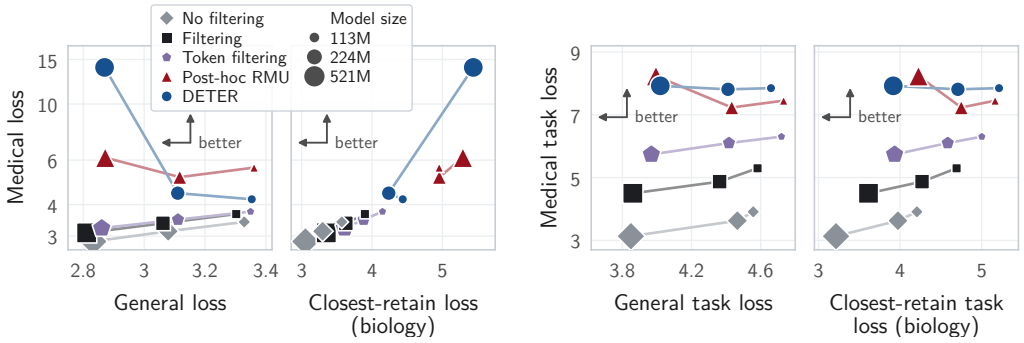


Figure 2: **At larger scale, suppression separates medicine more strongly from retained capabilities.** Each panel plots medical loss against retained loss before fine-tuning. The first two panels report corpus loss and the last two task loss. For corpus loss, medical, general, and closest-retain (biology) correspond to PubMed, retained FineWeb-Edu, and bioRxiv; for task loss, they correspond to the medical benchmarks, MMLU STEM and non-STEM, and MMLU biology. Larger markers denote larger models. Upward means stronger suppression of medicine; leftward means better preservation of retained capabilities. DETER denotes suppression during pretraining.

knowledge on MedQuAD (Ben Abacha & Demner-Fushman, 2019), MedMCQA (Pal et al., 2022), MedQA-USMLE (Jin et al., 2021), and the medicine subsets of MMLU (Hendrycks et al., 2021); biology knowledge on its biology subsets; and general knowledge on its STEM and non-STEM subsets. We report the negative log-likelihood of the correct answer text conditioned on the question (task loss; Bhagia et al., 2025). This continuous metric provides a more informative signal at small model scales, where discrete accuracy can remain near chance (Magnusson et al., 2025). Appendix D.3 details the prompting and loss normalization.

Adversarial fine-tuning. Following Rathi & Radford (2026), we fine-tune models on the PubMed section of the Common Pile and evaluate on its held-out PubMed corpus loss. We report the number of PubMed fine-tuning tokens required to reach the PubMed corpus loss of NO FILTERING, together with this quantity normalized by the corresponding number required for FILTERING at the same model size. The normalized value quantifies the additional fine-tuning required beyond filtering alone. For each model, the adversary uses the learning rate and schedule that reach this threshold with the fewest fine-tuning tokens (Appendix D.4).

3.3 RESULTS

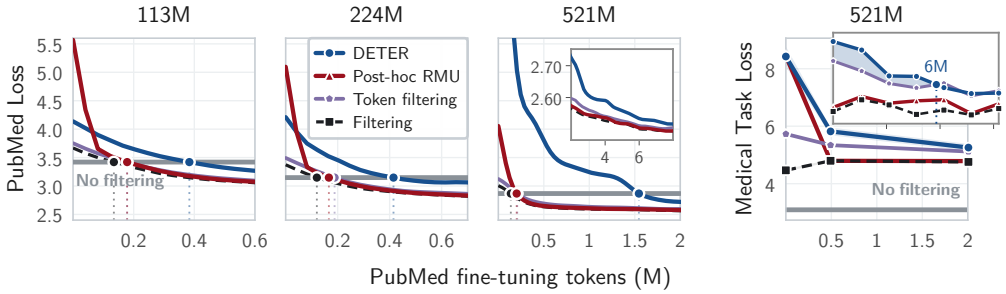


Figure 3: **Suppression during pretraining slows target-domain learning under fine-tuning on both medical corpus and task loss.** We fine-tune each model on PubMed and report held-out PubMed loss together with medical task loss, averaged over MedQuAD, MedMCQA, MedQA-USMLE, and MMLU medicine. The horizontal line marks the corresponding loss of NO FILTERING; for PubMed, crossing this line defines the fine-tuning budget reported in Table 1. The insets extend the 521M curves to 8M fine-tuning tokens. DETER denotes suppression during pretraining.

Suppression degrades medical capability beyond the documents it is applied to. Suppression is applied only to a subset of the medical documents in the pretraining corpus, yet before any fine-tuning the pretrained model is worse at medicine more broadly: loss increases on held-out PubMed text and on the correct answers of medical benchmarks (Figure 2). The damage is concentrated on

Table 1: **Suppression during pretraining requires substantially more target-domain fine-tuning than filtering, token filtering or post-hoc suppression, with little change in general loss.** General loss is measured on held-out retained FineWeb-Edu; medical task loss is averaged over MedQuAD, MedMCQA, MedQA-USMLE, and MMLU medicine; general task loss is measured on MMLU STEM and non-STEM; and closest-retain task loss on MMLU biology. All are reported relative to NO FILTERING. The last two columns give the number of PubMed fine-tuning tokens needed to reach the PubMed loss of NO FILTERING, and this number normalized by the corresponding value for FILTERING at the same model size. We average over three fine-tuning seeds. DETER denotes suppression during pretraining.

		Before medical fine-tuning on PubMed				After medical fine-tuning	
Size	Method	General loss ↓	General task loss ↓	Closest-retain task loss ↓	Medical task loss ↑	Tokens (M) ↑	Tokens ÷ FILTERING ↑
113M	FILTERING	−0.03	+0.03	+0.49	+1.39	0.13	1.00×
	TOKEN FILTERING	+0.02	+0.16	+0.80	+2.39	0.16	1.26×
	RMU	+0.03	+0.18	+0.96	+3.54	0.17	1.34×
	DETER	+0.02	+0.10	+1.01	+3.94	**0.39**	**3.05**×
224M	FILTERING	−0.02	−0.10	+0.30	+1.24	0.12	1.00×
	TOKEN FILTERING	+0.03	−0.05	+0.61	+2.47	0.18	1.51×
	RMU	+0.04	−0.03	+0.78	+3.61	0.17	1.43×
	DETER	+0.03	−0.05	+0.73	+4.19	**0.47**	**4.07**×
521M	FILTERING	−0.02	+0.01	+0.40	+1.37	0.12	1.00×
	TOKEN FILTERING	+0.03	+0.12	+0.72	+2.61	0.18	1.50×
	RMU	+0.04	+0.14	+1.01	+5.10	0.16	1.40×
	DETER	+0.04	+0.17	+0.70	+4.79	**1.58**	**13.53**×

medicine and collateral damage is limited. Biology, the closest retain domain, is affected less than medicine, as expected given its overlap with medical text, and general-domain evaluations far less still (Table 1, Appendix Figure 11).

With suppression during pretraining, unwanted knowledge is harder to acquire through adversarial fine-tuning. Under adversarial fine-tuning on PubMed, suppressed models require substantially more compute than filtering alone or post-hoc RMU to reach the NO FILTERING PubMed loss; we match RMU in general loss and target-token budget (Appendix D.1). Suppression also requires substantially more fine-tuning than token filtering, with the gap widening sharply at 521M (Table 1, Figure 4). For RMU, medical loss falls steeply early in fine-tuning and approaches the filtering curve, consistent with prior observations for post-hoc unlearning (Rathi & Radford, 2026; Shilov et al., 2025) (Figure 3). With suppression throughout pretraining, medical loss instead declines more gradually and remains above filtering throughout the attacks we run. For the 521M model, we extend the attack to 8M PubMed tokens (0.08% of its pretraining tokens), well beyond the point at which it reaches the NO FILTERING loss. On downstream medical tasks, the suppressed model remains above FILTERING even at the end of the attack, and does not converge with token filtering until roughly 6M fine-tuning tokens (Figure 3).

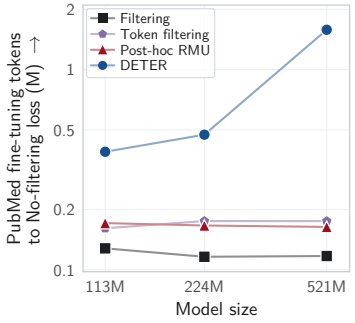


Figure 4: **Suppression during pretraining requires more target-domain fine-tuning than filtering, token filtering or post-hoc RMU, with the largest gap at 521M.** We report the number of PubMed fine-tuning tokens required to reach the PubMed loss of NO FILTERING at each model size. Higher values mean that more fine-tuning is needed to reach the same medical loss. We average over three fine-tuning seeds. DETER denotes suppression during pretraining.

Suppression during pretraining shows scaling behavior. The fine-tuning compute needed to undo suppression more than quadruples from our smaller to our larger model, and the gap relative to token filtering widens with scale (Table 1, Figure 4). At 521M, suppression requires 13.53× the fine-tuning of FILTERING (1.58M vs. 0.12M PubMed tokens). Post-hoc unlearning shows no such trend: RMU stays close to filtering at every size, so at the largest scale it requires nearly ten times less fine-tuning than suppression. The added resistance remains concentrated on medicine: at the two larger scales, suppression produces a larger separation between medical and retained loss than the other interventions we test (Figure 2).

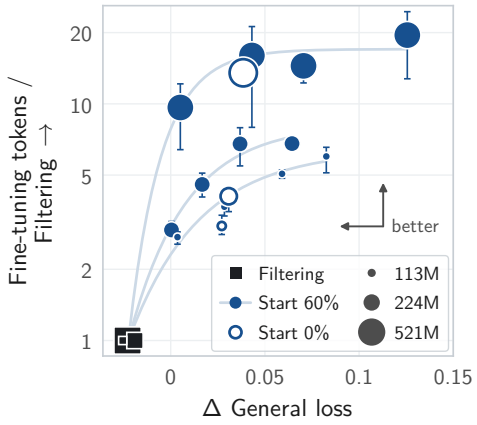


Figure 5: **At larger scale, suppression requires more target-domain fine-tuning for the same increase in general loss.** General loss is measured on held-out retained FineWeb-Edu, relative to NO FILTERING. Fine-tuning tokens / FILTERING is the number of PubMed fine-tuning tokens required to reach the PubMed loss of NO FILTERING, normalized by the corresponding value for FILTERING at the same model size. Points along each curve vary the suppression strength c. Points farther toward the upper left give a better trade-off.

Starting suppression earlier in pretraining becomes important for selective suppression at scale. Following Sam et al. (2026), we vary the onset of suppression over 0%, 20%, and 60% of pretraining. We keep the number of suppression steps fixed by shortening the interval P for later starts and choosing c to approximately match general loss (Appendix B.4). Across start times, later ones yield higher target-domain fine-tuning difficulty (Table 2). The shorter interval alone does not account for the difference: using it from the beginning of pretraining does not reproduce the resistance of a late start (Table 15). As models scale, however, earlier suppression becomes increasingly favorable for retained capabilities while giving up less of the fine-tuning difficulty achieved by a late start. At 521M, starting from 0% yields lower general and biology task loss than starting at 60%, yet retains about 85% of its normalized fine-tuning difficulty. The same scaling behavior is visible in Figure 5: the 0%-start run lies below the late-start frontier at smaller scales, but approaches it at 521M. This suggests that larger models become amenable to effective suppression earlier in their pretraining trajectory. One possible explanation is that, early in pretraining, the representations supporting medical predictions are not yet sufficiently differentiated from those supporting retained capabilities. Gradient ascent on medical data may therefore disrupt features shared across tasks rather than selectively impair medical capability. Smaller models may require more of their pretraining trajectory to develop representations that permit selective suppression, whereas larger models may reach this regime earlier.

Scaling improves the fine-tuning-difficulty–general-loss frontier. Suppression strength controls the trade-off between resistance to target-domain fine-tuning and retained performance (Figure 5). The coefficient c multiplies the learning rate of each suppression step (Eq. 1), and therefore controls the strength of each suppression update. Each curve starts suppression at 60% of pretraining with $P = 20$ and varies only c. For each model size, increasing c initially makes the model require more medical fine-tuning on PubMed to reach the NO FILTERING medical loss, with a relatively small increase in general loss. At larger c, this benefit saturates: additional suppression causes substantially more general loss with little further increase in fine-tuning difficulty. As model size increases, this trade-off improves. Larger models can require more target-domain fine-tuning for the same increase in general loss, and can achieve the same fine-tuning difficulty with less general degradation. The highest fine-tuning difficulty reached along the curve also increases with model size.

Table 2: **Starting suppression earlier becomes more effective as model size increases.** We compare suppression onsets while keeping the number of suppression updates fixed and approximately matching general loss. General loss is measured on held-out retained FineWeb-Edu; general task loss is measured on MMLU STEM and non-STEM; and closest-retain task loss on MMLU biology. All are reported relative to NO FILTERING and values near zero suggest better preserved general performance. The final column gives the number of PubMed fine-tuning tokens needed to reach the PubMed loss of NO FILTERING, normalized by the corresponding value for FILTERING at the same model size. We average over three fine-tuning seeds.

		Before medical fine-tuning on PubMed			After medical fine-tuning
Size	Start	General loss ↓	General task loss ↓	Closest retain task loss ↓	Tokens ÷ FILTERING ↑
113M	0%	+0.02	+0.10	+1.01	3.05×
	20%	+0.02	+0.09	+0.90	3.01×
	60%	+0.03	+0.07	+1.18	**3.67**×
224M	0%	+0.03	−0.05	+0.73	4.07×
	20%	+0.03	−0.08	+0.72	6.11×
	60%	+0.04	+0.03	+0.96	**6.78**×
521M	0%	+0.04	+0.17	+0.70	13.53×
	60%	+0.04	+0.22	+1.02	**16.01**×

4 Discussion and Limitations

Choice of suppression loss. At 113M with AdamW, we compare gradient ascent with six other choices of $\mathcal{L}_F$. Two carry no target distribution: (1) *truncated gradient ascent* stops the ascent at each token whose probability has fallen below that of a uniform prediction; and (2) *local entropy*, the local loss of PALU (Zhai et al., 2026), pulls the 5,000 largest logits toward the mean logit at the position. Two take as target a distribution in which the observed token, or every token, has the probability of a uniform prediction: (3) *maximum entropy*, which maximises the entropy of the model's next-token distribution; and (4) *uniform-target self-distillation*, the forget term of Unilogit (Vasilev et al., 2025). Two use a weaker predictor of the target domain as the target distribution: (5) *past-self distillation* distils the model's next-token distribution toward that of $\theta_{t_\star}$, an early checkpoint of the same model with $t_\star \leq t_{\text{start}}$; and (6) *small-model distillation* distils toward a 61M model pretrained on $\mathcal{D}_\rho$ and held fixed throughout pretraining (see Appendix B.3 for full definitions). Gradient ascent gives the strongest resistance–general-loss trade-off (Figure 6). We extend the best-performing objectives to 224M, where the advantage of gradient ascent appears to persist. Gradient ascent lowers the likelihood of observed target-domain tokens without specifying a replacement distribution. Ordinary pretraining, meanwhile, continues to provide a positive learning signal on retained data. One possible explanation for its effectiveness is that the two objectives serve complementary roles: suppression opposes target-domain predictions, while pretraining supports useful predictions elsewhere. More prescriptive objectives add assumptions about what the model should predict on target-domain examples, which appear not to translate into a better trade-off.

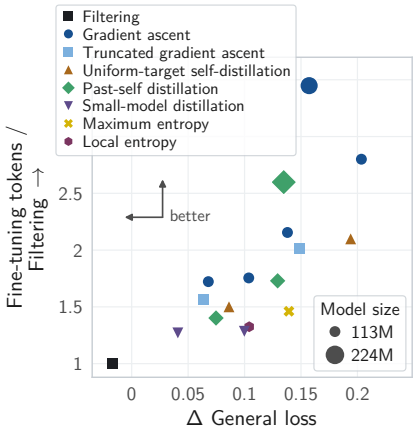


Figure 6: **Gradient ascent gives the strongest general-loss–fine-tuning difficulty trade-off among the suppression objectives tested.** General loss is measured on held-out retained FineWeb-Edu relative to No filtering. Fine-tuning tokens / Filtering is the number of PubMed fine-tuning tokens required to reach the PubMed loss of No filtering, normalized by the corresponding value for Filtering at the same model size. Points farther toward the upper left give a better trade-off. Points for the same objective vary the suppression strength c.

Table 3: **Separate optimizer states require more target-domain fine-tuning and substantially reduce spillover into the closest retain domain.** General loss is measured on held-out retained FineWeb-Edu, medical loss is measured on PubMed, and closest-retain loss on non-medical bioRxiv subjects. All losses are reported relative to No filtering. The final column gives the number of PubMed fine-tuning tokens needed to reach the PubMed loss of No filtering, normalized by the corresponding value for Filtering at the same model size.

	Before medical fine-tuning on PubMed			After medical fine-tuning
	General loss ↓	Closest retain loss ↓	Medical loss ↑	Fine-tuning tokens ÷ Filtering ↑
Muon, 113M				
Separate	−0.02	+0.57	+0.47	**1.66**×
Shared	−0.01	+25.60	+10.99	1.62×
AdamW, 113M				
Separate	+0.07	+0.75	+0.75	**1.93**×
Shared	+0.05	+14.13	+13.83	1.79×
AdamW, 224M				
Separate	+0.09	+2.02	+8.11	**3.35**×
Shared	+0.09	+54.10	+25.23	2.47×

How should suppression interact with pretraining? All models in our main experiments maintain separate optimizer statistics for pretraining and suppression (Section 2). Sharing statistics makes suppression appear stronger at the end of pretraining, but nearly all of the additional medical loss disappears within the first few adversarial fine-tuning steps. It also causes much greater spillover into biology despite comparable general loss. Separate statistics instead yield greater resistance with both Muon and AdamW, with a more pronounced advantage in the larger model (Table 3). Sharing statistics couples the objectives beyond their effects on model parameters: pretraining gradients influence the optimizer history used for suppression, and suppression gradients influence the history used for pretraining. Separate statistics keep those histories distinct while allowing both objectives to shape the same model. For AdamW, the distinction extends beyond momentum: at

113M keeping separate first moments while sharing second moments severely degrades general performance. Independent momentum histories alone therefore do not reproduce the benefits of fully separate optimizer states.

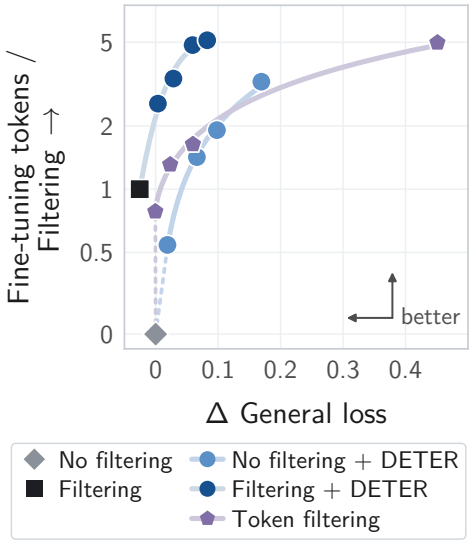


Figure 7: **Suppression increases the fine-tuning required even without filtering, and is strongest when combined with document filtering.** We plot general loss relative to NO FILTERING against the PubMed fine-tuning tokens required to reach the loss of NO FILTERING, normalized by FILTERING. For suppression with and without document filtering, connected points vary the suppression strength c; for TOKEN FILTERING, they vary the filtering rate. NO FILTERING and FILTERING are shown as reference points. Points farther toward the upper left are better. DETER denotes suppression during pretraining.

Does suppression require filtering? Suppression builds resistance even without filtering the pretraining corpus. Over part of the evaluated range, suppression on the unfiltered stream approaches the resistance–general-loss trade-off of token filtering (Figure 7), despite medical content remaining available for ordinary pretraining. Filtering is therefore not a prerequisite for suppression to be effective. Nevertheless, combining suppression with document filtering substantially improves the trade-off: the combination achieves high resistance at much lower general loss than either token filtering alone or suppression on the unfiltered stream. These results show that suppression can act both independently of and alongside data filtering. They also motivate combining suppression with token-level filtering, a setting we do not evaluate here.

At what granularity should suppression act? At matched general loss, restricting suppression to medical tokens selected by a token-level mask (Appendix C.3) reduces spillover to biology, but lowers resistance relative to suppressing the full target-domain document (Table 4). The comparison is made at a scale where the suppression steps can draw entirely from the most medical portion of the filtered-out data, where target-domain content is already highly concentrated. The weaker resistance of token-level suppression may partly reflect noise in the token-level mask, rather than an inherent disadvantage of operating at finer granularity. At larger pretraining scales, the additional suppression updates could instead be used to broaden coverage of the target domain by drawing on more of the filtered-out corpus. Because these additional documents contain a greater mixture of medical and retain-domain content, token-level selection may prove useful in that regime.

Which data should suppression use? The data most useful for suppression need not be identical to the data that should be removed from pretraining. We construct $\mathcal{D}_F$ from the highest-scoring FineWeb-Edu documents accounting for 5% of corpus tokens, within the broader 20% removed by filtering. This uses different selection thresholds for excluding potential sources of target-domain learning and supplying negative training examples. A high classifier score, however, does not establish that a document provides the most useful suppression signal or that the selected set covers the full target domain. A narrow selection may leave some subdomains underrepresented, whereas a broader selection may introduce more retain-domain content. The observed transfer to PubMed and medical benchmarks demonstrates effects beyond the suppression documents, but does not establish that the current selection is optimal. Comparing selection thresholds, subdomain coverage, and alternative sources of suppression data would clarify their effects on resistance and collateral damage.

How do classifier errors affect suppression? Classifier errors affect both the pretraining stream and the suppression signal. False negatives at the filtering threshold leave target-domain documents available for ordinary pretraining. Suppression on other examples may counteract this learning, but only insofar as its effects generalize beyond the suppression set. False positives remove potentially

useful retain-domain documents, as in filtering alone. When these documents also enter the suppression set, however, gradient ascent can additionally discourage predictions on retain-domain content, creating a further route to collateral damage. Documents removed by filtering but not selected for suppression are only omitted, rather than actively trained against. Suppression therefore changes the consequences of classifier errors rather than eliminating them: classifier quality determines not only what the model learns from, but also what it is trained against.

Limitations. Our experiments use medical knowledge as a proxy target domain in models of up to 521M parameters, and therefore do not establish that the observed scaling trends extend to genuinely hazardous capabilities or to other domains and bigger model scales. We do not systematically vary classifier quality or suppression-set construction, leaving sensitivity to false positives, missed subdomains, and the source and diversity of suppression data unresolved. The same classifier also defines the FineWeb-Edu evaluation splits, which may share blind spots with training-data selection; the external corpora and benchmarks provide complementary but not exhaustive coverage. Furthermore, recovery is measured through held-out PubMed loss under fine-tuning attacks on models pretrained with a single seed; we do not exhaustively evaluate adversarial modifications. Every method reaches the NO FILTERING baseline's PubMed loss within the evaluated attack budget. In absolute terms, these budgets are small: at 521M, filtering reaches the NO FILTERING loss after 0.12M PubMed tokens and suppression after 1.58M, or 0.015% of its pretraining tokens. Our findings therefore establish that more target-domain fine-tuning is required under the attacks and the model sizes we test.

Table 4: **Token-level suppression makes capability unlearning more concentrated, but document-level suppression makes target capability harder to acquire with fine-tuning.** All losses are relative to NO FILTERING; retain/target is the increase on bioRxiv divided by that on PubMed. The final column gives the number of PubMed fine-tuning tokens needed to reach the PubMed loss of NO FILTERING, normalized by the corresponding value for FILTERING at the same model size. We average over three fine-tuning seeds. Token-level suppression applies the loss only to the tokens that a token-level probe labels medical; the confident variant applies it only to the tokens the probe scores highest.

	General loss ↓	Retain / target ↓	Tokens ÷ FILTERING ↑
`AdamW`, 113M			
FILTERING	−0.02	1.41	1.00×
Token, confident	+0.07	**1.01**	1.81×
Token	+0.07	1.03	1.96×
Document	+0.07	1.08	**2.00**×
`Muon`, 113M			
FILTERING	−0.03	1.29	1.00×
Token, confident	+0.02	**1.04**	2.43×
Token	+0.03	1.12	2.94×
Document	+0.02	1.11	**3.05**×

Future work. Several directions would extend the present study. Broader comparisons could evaluate SGTM (Shilov et al., 2025) and TAR (Tamirisa et al., 2025), trace filtering and suppression trade-offs more extensively, account for the additional suppression compute when comparing retained utility, and test whether using different optimizers for pretraining and suppression increases the amount of target-domain fine-tuning required while preserving retained performance, as explored for separate objectives in post-hoc unlearning (Zhong et al., 2025). Extending the experiments to larger models and additional target domains would test how broadly the observed scaling behavior holds and enable more controlled study of the factors underlying it. Finally, representation-level analyses could investigate why suppression during pretraining makes target-domain capability slower to acquire through fine-tuning, including whether interventions differ in residual target knowledge or in the organization of target and retained representations.

5 CONCLUSION

The effectiveness of simple gradient ascent motivates studying the fundamentals of suppression within pretraining. At the largest scale studied, earlier suppression improves the balance between resistance and general loss, highlighting the importance of how suppression interacts with ordinary learning. Target-domain learning through fine-tuning remains possible; the goal is to raise its cost while preserving useful capabilities. Establishing how to achieve that balance across scales and domains should be part of understanding pretraining itself, rather than left entirely to the study of post-hoc unlearning.

AI USE STATEMENT

We used AI tools to assist with code implementation and to provide feedback on research methodology and experiments. We additionally used these tools for manuscript editing and literature search.

ETHICS STATEMENT

REPRODUCIBILITY STATEMENT

Section 2 describes the suppression procedure. Appendix B provides model architectures, optimization details, and the full training procedure; Appendix B.4 reports the suppression hyperparameters used in each experiment; Appendix C describes data construction and target-domain selection; and Appendix D gives baseline implementations and evaluation details, including the adversarial fine-tuning procedure and hyperparameter search. Code is available at `https://anonymous.4open.science/r/DETER-46B7/`.

REFERENCES

Nora Belrose, Quintin Pope, Lucia Quirke, Alex Mallen, and Xiaoli Fern. Neural networks learn statistics of increasing complexity. In *Proceedings of the International Conference on Machine Learning*, 2024.

Asma Ben Abacha and Dina Demner-Fushman. A question-entailment approach to question answering. *BMC Bioinformatics*, 2019.

Akshita Bhagia, Jiacheng Liu, Alexander Wettig, David Heineman, Oyvind Tafjord, Ananya Harsh Jha, Luca Soldaini, Noah A. Smith, Dirk Groeneveld, Pang Wei Koh, Jesse Dodge, and Hannaneh Hajishirzi. Establishing task scaling laws via compute-efficient model ladders. In *Conference on Language Modeling*, 2025.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. Characterizing learning curves during language model pre-training: Learning, forgetting, and stability. *Transactions of the Association for Computational Linguistics*, 2024.

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.

Alex Cloud, Jacob Goldman-Wetzler, Evžen Wybitul, Joseph Miller, and Alexander Matt Turner. Gradient routing: Masking gradients to localize computation in neural networks. *arXiv preprint arXiv:2410.04332*, 2024.

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*, 2025.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Peter Henderson, Eric Mitchell, Christopher Manning, Dan Jurafsky, and Chelsea Finn. Self-destructing models: Increasing the costs of harmful dual uses of foundation models. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2023.

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*, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy,

Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, 2022.

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 Annual Meeting of the Association for Computational Linguistics*, 2023.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 2021.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. `https://kellerjordan.github.io/posts/muon/`, 2024.

Nikhil Kandpal, Brian Lester, Colin Raffel, Sebastian Majstorovic, Stella Biderman, Baber Abbasi, Luca Soldaini, Enrico Shippole, A. Feder Cooper, Aviya Skowron, John Kirchenbauer, Shayne Longpre, Lintang Sutawika, Alon Albalak, Zhenlin Xu, Guilherme Penedo, Loubna Ben Allal, Elie Bakouch, John David Pressman, Honglu Fan, Dashiell Stander, Guangyu Song, Aaron Gokaslan, Tom Goldstein, Brian R. Bartoldson, Bhavya Kailkhura, and Tyler Murray. The Common Pile v0.1: An 8TB dataset of public domain and openly licensed text. In *Advances in Neural Information Processing Systems*, 2025.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. Pretraining language models with human preferences. In *Proceedings of the International Conference on Machine Learning*, 2023.

Bruce W. Lee, Addie Foote, Alex Infanger, Leni Shor, Harish Kamath, Jacob Goldman-Wetzler, Bryce Woodworth, Alex Cloud, and Alexander Matt Turner. Distillation robustifies unlearning. In *Advances in Neural Information Processing Systems*, 2025.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B. Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam Alfred Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Proceedings of the International Conference on Machine Learning*, 2024.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D. Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, Noah A. Smith, Pang Wei Koh, and Jesse Dodge. DataDecide: How to predict best pretraining data with small experiments. In *Proceedings of the International Conference on Machine Learning*, 2025.

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*, 2026.

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

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, 2024.

Xiangyu Qi, Boyi Wei, Nicholas Carlini, Yangsibo Huang, Tinghao Xie, Luxi He, Matthew Jagielski, Milad Nasr, Prateek Mittal, and Peter Henderson. On evaluating the durability of safeguards for open-weight LLMs. In *International Conference on Learning Representations*, 2025.

Xingyu Qu, Peigeng Huang, and Samuel Horváth. Can Muon fine-tune Adam-pretrained models? In *Proceedings of the International Conference on Machine Learning*, 2026.

Neil Rathi and Alec Radford. Shaping capabilities with token-level data filtering. *arXiv preprint arXiv:2601.21571*, 2026.

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Hassan Sajjad, and Frank Rudzicz. Immunization against harmful fine-tuning attacks. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024.

Dylan Sam, Sachin Goyal, Pratyush Maini, Alexander Robey, and J. Zico Kolter. When should we introduce safety interventions during pretraining? *arXiv preprint arXiv:2601.07087*, 2026.

Igor Shilov, Alex Cloud, Aryo Pradipta Gema, Jacob Goldman-Wetzler, Nina Panickssery, Henry Sleight, Erik Jones, and Cem Anil. Beyond data filtering: Knowledge localization for capability removal in LLMs. *arXiv preprint arXiv:2512.05648*, 2025.

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.

Stefan Vasilev, Christian Herold, Baohao Liao, Seyyed Hadi Hashemi, Shahram Khadivi, and Christof Monz. Unilogit: Robust machine unlearning for LLMs using uniform-target self-distillation. In *Findings of the Association for Computational Linguistics: ACL*, 2025.

Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems*, 2021.

Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-aware data selection for efficient pretraining with data influence models. In *Advances in Neural Information Processing Systems*, 2024.

Naixin Zhai, Pengyang Shao, Binbin Zheng, Yonghui Yang, Fei Shen, Long Bai, and Xun Yang. Maximizing local entropy where it matters: Prefix-aware localized LLM unlearning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2026.

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

Xuyang Zhong, Haochen Luo, and Chen Liu. DualOptim: Enhancing efficacy and stability in machine unlearning with dual optimizers. In *Advances in Neural Information Processing Systems*, 2025.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *Advances in Neural Information Processing Systems*, 2024.

A Related work

Preference-informed pretraining. Korbak et al. (2023) incorporate human preferences into pretraining to steer model generations away from undesirable behavior, finding conditional training particularly effective. Their results show that safety-relevant signals can be more effective when introduced during pretraining than through subsequent fine-tuning. We study a different objective: suppressing the acquisition of domain knowledge and increasing the fine-tuning required to acquire it later.

Pretraining data filtering. Filtering suppresses capabilities by removing their training signal altogether. O'Brien et al. (2026) show that filtering biothreat-related data can remain effective under extensive adversarial fine-tuning, and Rathi & Radford (2026) find that token-level filtering improves the retain–forget trade-off, strengthens with scale, and remains effective under label noise given sufficient pretraining compute. We use filtering as our primary baseline and ask whether the excluded target-domain data can instead be used to make the resulting model harder to fine-tune on that domain.

Intervention timing. The effect of a pretraining intervention can depend on when it begins. For behavior-shaping safety interventions, Sam et al. (2026) find that earlier introduction generally improves robustness to subsequent fine-tuning; for capability filtering, Rathi & Radford (2026) find that delaying loss masking weakens suppression. We therefore treat suppression start time as an experimental variable.

Localizing capabilities during training. Gradient Routing and Selective GradienT Masking (SGTM) control where target-domain information is learned by restricting its gradients to designated parameters that can later be removed (Cloud et al., 2024; Shilov et al., 2025). SGTM is robust to adversarial fine-tuning and can outperform filtering when target-domain labels are imperfect. Our approach uses the shared model parameters throughout, testing whether periodic suppression updates can make target knowledge harder to acquire without reserving or removing parameters.

Post-hoc unlearning. Post-hoc unlearning modifies a model after pretraining using objectives on its outputs (Jang et al., 2023; Zhang et al., 2024; Fan et al., 2025; Zhai et al., 2026; Vasilev et al., 2025) or representations (Li et al., 2024). Such interventions can be rapidly reversed by fine-tuning (Lee et al., 2025). Optimizer choice and state can also affect post-hoc unlearning: Zhong et al. (2025) use a separate adaptive optimizer for forgetting and maintain separate optimizer states for forget and retain updates. In our setting, pretraining and suppression use the same optimizer types for the same parameter groups, and we study whether their optimizer states should be shared. We compare against post-hoc RMU to test whether applying suppression during the pretraining trajectory changes the subsequent cost of acquiring target-domain knowledge.

Tamper resistance. Tamper-resistant training explicitly optimizes models against simulated fine-tuning adversaries (Henderson et al., 2023; Tamirisa et al., 2025). Its evaluation depends critically on attack strength: Qi et al. (2025) show that stronger fine-tuning configurations can substantially reduce previously reported durability. Following Rathi & Radford (2026), we therefore tune the adversarial fine-tuning learning rate and measure the target-domain fine-tuning required to reach the corresponding No filtering model.

B Implementation details

B.1 Models and pretraining

We follow the architecture and scale ladder of Rathi & Radford (2026): decoder-only transformers with rotary position embeddings, squared-ReLU activations, pre-RMSNorm, a width-to-depth ratio of 64, and μP transfer from a base width of 512. All models use a context length of 2,048 tokens and an effective batch size of 327,680 tokens. Following Rathi & Radford (2026), we use a peak learning rate of 3×10^{-3} at the base width for the weight matrices inside the transformer blocks and transfer it across scales with μP. The learning-rate schedule uses linear warmup over the first 10% of training

followed by cosine decay to 10% of the peak rate. Each model is trained Chinchilla-optimally, on 20 tokens per parameter (Hoffmann et al., 2022).

We pretrain models under two optimizer configurations, which we refer to by their main optimizer. Under `Muon`, we update the weight matrices inside the transformer blocks with `Muon` and the embedding and normalization parameters with `AdamW`. We rescale each orthogonalized update to match the root-mean-square of an `AdamW` update (Liu et al., 2025), and orthogonalize the query, key, and value projections separately. Under `AdamW`, we update all parameters with `AdamW`.

The unfiltered baseline, filtered baseline, and suppression-pretrained models use identical architecture and optimization settings and differ only in the suppression configuration described in Section 2. Table 5 reports the architecture at each scale.

Table 5: Architecture for each model scale.

Model	Layers	Heads	Embed dim
61M	7	8	448
113M	10	10	640
224M	14	14	896
521M	20	10	1,280

B.2 Suppression steps

Algorithm 1 gives the training procedure. Pretraining and suppression steps apply the same optimizer to the same parameters θ, but each keeps its own gradient statistics. s and s_F hold what the optimizer accumulates from past gradients: for momentum and matrix-based methods (`SGD` with momentum, `Muon`), one momentum buffer per parameter; for coordinate-wise adaptive methods (`Adam`, `AdamW`), the first- and second-moment estimates. s is updated only by gradients of ℓ_{pre}, and s_F only by gradients of $\mathcal{L}_F$ as in Eq. (1) (Figure 9).

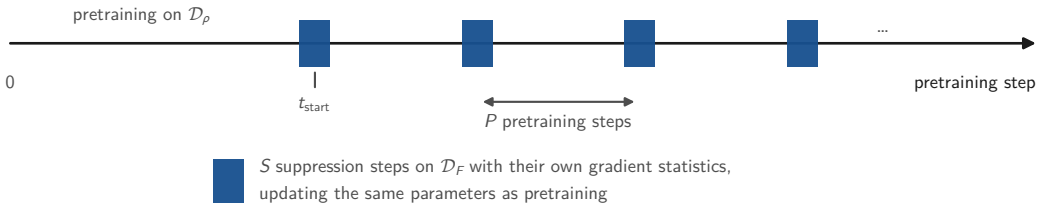


Figure 8: Schedule of suppression steps during pretraining.

Efficiency analysis. When using the same batch size, a suppression update requires approximately the same compute as a pretraining update: one forward and one backward pass. During the interval in which suppression is active, the additional compute is therefore approximately S/P of pretraining compute. Losses requiring an additional forward pass through a reference model of the same size increase this estimate to $4S/(3P)$. For $S = 1$ and $P = 50$, these correspond to approximately 2% and 2.7%, respectively. A delayed suppression start reduces the overhead averaged over the full pretraining. The primary memory overhead stems from the second set of moment estimates that suppression maintains. If model parameters require M storage, `AdamW` stores approximately $4M$ (parameters, gradients, and 2 EMA states). The suppression steps store additionally $2M$ for the two EMA states, leading to $\approx 50\%$ memory overhead. With `Muon`, the weight matrices inside the transformer blocks carry a single momentum buffer rather than two moment estimates, halving their contribution to this overhead, which tends toward $\approx 33\%$ as transformer parameters dominate at larger scales. Losses requiring a reference model need additional memory for the reference model's parameters and gradients.

Algorithm 1 Suppression during pretraining.

Input: $\mathcal{D}_\rho$: filtered corpus; $\mathcal{D}_F \subseteq \mathcal{D}$: target-domain documents
Input: ℓ_{pre}: next-token loss; $\mathcal{L}_F$: suppression objective
Input: T: pretraining steps; t_{start}: first suppression step; P: interval; S: suppression steps
Input: c: suppression learning-rate multiplier
Input: Opt: optimizer, with learning-rate schedule η_t and weight decay λ
Input: s, s_F: gradient statistics of pretraining and of suppression
Input: θ_0: initial parameters

$\theta \leftarrow \theta_0, \quad s, s_F \leftarrow 0$
for $t = 0$ **to** $T - 1$ **do**
  if $t \geq t_{\text{start}}$ **and** $t \bmod P = 0$ **then** *# Suppression step*
    for $j = 1$ **to** S **do**
      Sample a batch $B_F \sim \mathcal{D}_F$
      Compute the suppression loss $\mathcal{L} \leftarrow \mathcal{L}_F(\theta; B_F)$
      Update parameters $(\theta, s_F) \leftarrow \text{Opt}(\theta, \nabla_\theta \mathcal{L}, s_F; c\,\eta_t, \lambda)$
    end for
  end if
  Sample a batch $B \sim \mathcal{D}_\rho$ *# Pretraining step*
  Update parameters $(\theta, s) \leftarrow \text{Opt}(\theta, \nabla_\theta \ell_{\text{pre}}(\theta; B), s; \eta_t, \lambda)$
end for
Return parameters θ_T

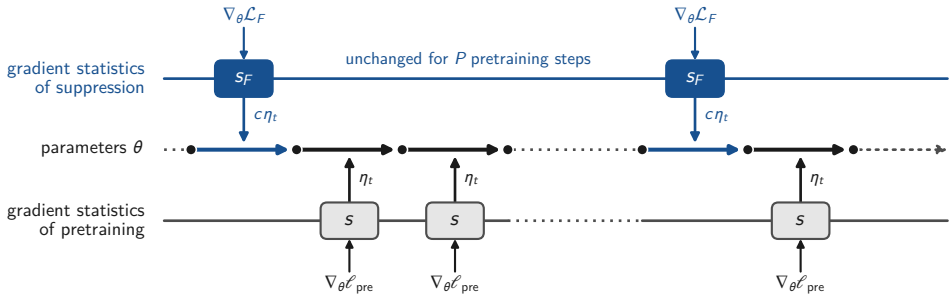


Figure 9: Optimization states of pretrainer (learner) and suppressor (unlearner) throughout pretraining with interleaved suppression.

B.3 Suppression losses

We specify the per-token suppression loss ℓ in Eq. (1) for each loss considered in Section 3. All losses below are minimized in the suppression update; in particular, minimizing $\log p_\theta(x_i \mid x_{<i})$ corresponds to gradient ascent on the next-token cross-entropy. Let $z_\theta(x_{<i}) \in \mathbb{R}^{|V|}$ denote the logits at position i, $p_\theta(\cdot \mid x_{<i})$ their softmax, and x_i the observed token. Teacher and target distributions are treated as fixed during each suppression update.

- **Gradient ascent.** Following Jang et al. (2023),

$$\ell(\theta; x, i) = \log p_\theta(x_i \mid x_{<i}).$$

- **Truncated gradient ascent.** We stop suppressing a token once its probability falls below that of a uniform prediction:

$$\ell(\theta; x, i) = \max\{\log p_\theta(x_i \mid x_{<i}), -\log |V|\}\,.$$

- **Local entropy.** Following the local-entropy objective of PALU (Zhai et al., 2026), we flatten the K largest logits toward the mean logit at each selected position:

$$\ell(\theta; x, i) = \frac{1}{K} \sum_{v \in V_{\text{top}}} \left(z_{\theta,v}(x_{<i}) - \bar{z}\right)^2, \qquad \bar{z} = \frac{1}{|V|} \sum_{v \in V} z_{\theta,v}(x_{<i}).$$

We use $K = 5{,}000$ and, unlike PALU, recompute V_{top} from the current model at each suppression step. The loss is evaluated at every position selected by m.

- **Maximum entropy.** We minimize the divergence from a uniform next-token distribution:

$$\ell(\theta; x, i) = \log |V| - H(p_\theta(\cdot \mid x_{<i})) = D_{\text{KL}}\big(p_\theta(\cdot \mid x_{<i}) \,\big\|\, \text{Unif}(V)\big)\,.$$

- **Uniform-target self-distillation.** We use the forget objective of Unilogit (Vasilev et al., 2025). At each suppression step, we construct a detached target from the current logits, changing only the logit of x_i so that its target probability equals $1/|V|$:

$$\ell(\theta; x, i) = D_{\text{KL}}\big(p_\theta(\cdot \mid x_{<i}) \,\big\|\, \text{softmax}(\tilde{z})\big)\,,$$

where

$$\tilde{z}_v = \begin{cases} \log \dfrac{\sum_{u \neq x_i} e^{z_{\theta,u}(x_{<i})}}{|V| - 1}, & v = x_i, \\ z_{\theta,v}(x_{<i}), & v \neq x_i. \end{cases}$$

The target $\tilde{z}$ is detached before computing the KL divergence.

- **Past-self distillation.** We distil toward a frozen checkpoint $\theta_{t_\star}$ of the same model, with $t_\star \leq t_{\text{start}}$:

$$\ell(\theta; x, i) = D_{\text{KL}}\big(p_\theta(\cdot \mid x_{<i}) \,\big\|\, p_{\theta_{t_\star}}(\cdot \mid x_{<i})\big)\,.$$

We use the checkpoint after 5% of pretraining as the teacher. Early in pretraining, language models predict mainly from low-order statistics such as token and n-gram frequencies (Belrose et al., 2024; Chang et al., 2024). Distilling toward such a teacher replaces target-domain predictions with generic ones that pretraining itself produces, rather than with the uninformative predictions of a teacher at initialization. At matched general loss, the 5% teacher gives higher resistance (Table 6). We do not tune the checkpoint further.

- **Small-model distillation.** We distil toward a frozen 61M model with the same vocabulary:

$$\ell(\theta; x, i) = D_{\text{KL}}\big(p_\theta(\cdot \mid x_{<i}) \,\big\|\, p_{\text{small}}(\cdot \mid x_{<i})\big)\,.$$

We consider three teachers: a model pretrained on $\mathcal{D}_\rho$ with $\rho = 20\%$; the same model trained with gradient-ascent suppression; and a model pretrained on $\mathcal{D}$ and subsequently unlearned with RMU (Li et al., 2024).

Table 6: Past-self distillation at 113M with `AdamW`, with the teacher at initialization (0%) and after 5% of pretraining. General log-perplexity over FILTERING (↓) and resistance on PubMed as a multiple of the compute against FILTERING (↑).

Teacher	General loss ↓	Resistance ↑
0%	+0.08	1.04
5%	+0.09	1.42

Retain loss. Unlearning methods typically add a retain loss, which keeps the model close to a frozen copy of itself on general data (Li et al., 2024; Vasilev et al., 2025; Zhai et al., 2026). In our setting, the retain loss is a KL divergence on general documents to the predictions of θ_t, the model just before the suppression update; when $S \geq 2$, it adds a forward pass of θ_t and a forward and backward pass at every suppression step after the first. Since we take one or very few suppression steps, and the pretraining steps between them protect general capabilities, we drop the retain loss to avoid its cost and let c and P control suppression (Table 7).

Table 7: Past-self distillation at 113M with AdamW with and without the retain loss. General log-perplexity over FILTERING (↓), resistance on PubMed as a multiple of the compute against FILTERING (↑), and compute per suppression in pretraining steps (↓).

Retain loss	S	General loss ↓	Resistance ↑	Compute ↓
No	1	+0.09	1.42	1
Yes	2	+0.09	1.43	$3\frac{1}{3}$
No	2	+0.13	1.51	2

B.4 SUPPRESSION HYPERPARAMETERS

Table 8 lists t_{start}, as a share of pretraining, P and c for every suppression-pretrained model. Every model uses $S = 1$ without a retain loss and pretrains on $\mathcal{D}_\rho$, except the NO FILTERING models, which pretrain on $\mathcal{D}$.

Table 8: Suppression hyperparameters.

	Model	Start	P	c	Reported in
Muon, gradient ascent, whole document					
Main models	113M, 224M, 521M	0%	50	5	Tables 1, 2, 4, 13, **??**; Figures 2–5, 11
Start time	113M	20%	40	4.5	Table 2
	113M	60%	20	3.8	Table 2
	113M	0%	20	2.2	Section 3
	224M	20%	40	4.2	Table 2
	224M	60%	20	3.8	Table 2
	521M	60%	20	3.0	Table 2
Suppression strength	113M	60%	20	2.5, 3.8, 5, 6	Figures 5, 7
	224M	60%	20	1.9, 2.9, 3.8, 4.8	Figure 5
	521M	60%	20	2, 3, 3.7, 5	Figure 5
Unfiltered corpus	113M	60%	20	3.3, 5.4, 6.3, 7.8	Figure 7
Token-level	113M	0%	50	5	Table 4
Separate and shared statistics	113M	5%	50	2	Table 3
AdamW, gradient ascent, whole document					
Main models	113M	5%	50	2	Tables 3, 4, 13
	224M	5%	50	3	Tables 3, 13
Shared statistics	113M	5%	50	2	Table 3
	224M	5%	50	4	Table 3
AdamW, token-level					
Gradient ascent	113M	5%	50	2, 2.5, 3, 4	Table 4; Figures 6, 10
	224M	5%	50	4	Figure 6
Truncated gradient ascent	113M	5%	50	2, 3	Figures 6, 10
Uniform-target self-distillation	113M	5%	50	2, 3	Figures 6, 10
Past-self distillation	113M	5%	50	2, 3	Figures 6, 10
	224M	5%	50	4	Figure 6
top 1,024 tokens	113M	5%	50	2	Tables 6, 7; Figure 10
teacher at initialisation	113M	5%	50	2	Table 6; Figure 10
rolling teacher	113M	5%	50	2	Figure 10
Small-model distillation	113M	5%	50	2, 3	Figures 6, 10
Maximum entropy	113M	5%	50	2	Figures 6, 10
Local entropy	113M	5%	50	2	Figures 6, 10

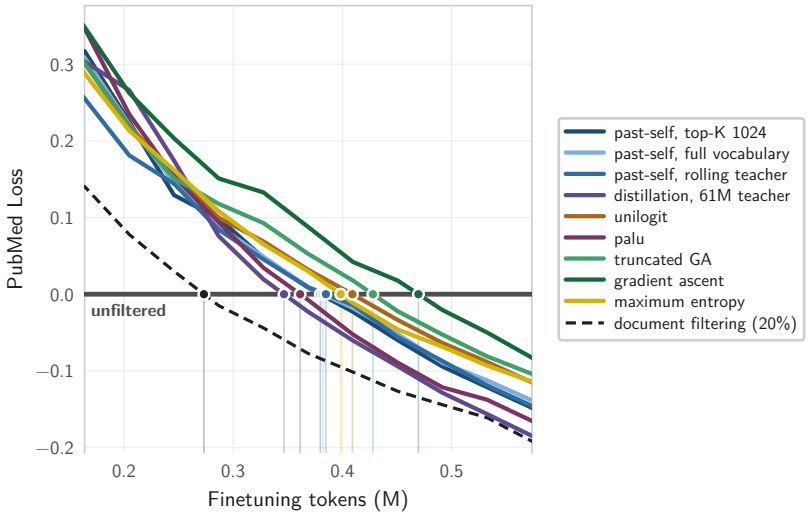


Figure 10: **Gradient ascent slows fine-tuning on PubMed the most among the suppression objectives.** We fine-tune each 113M `AdamW` model on PubMed and report held-out PubMed loss. The horizontal line marks the PubMed loss of NO FILTERING.

C DATA

C.1 TARGET DOMAIN

Category definition. We target medical content, following Rathi & Radford (2026). A document is medical when its subject is clinical information, symptoms, diagnoses or treatments; medical or pharmaceutical research or industry; medical devices and procedures; human anatomy and physiology; virology, immunology or disease in humans; neurology and neurological disorders; or medical genetics. It is not medical when its subject is colloquial, non-medical references to anatomy; cosmetic surgery; general non-medical biology or chemistry; animal behaviour and cognition; non-medical biochemistry or genetics; healthcare policy, administration or education; psychiatry, mental illness or psychology; wellness and meditation; public health and epidemiology; or pregnancy and childcare. The document annotator chooses one of five classes — medical, veterinary, molecular biology, organismal biology, other — and we take medical against the remaining four, which keeps animal medicine and both biologies off the medical label. Both lists appear verbatim in every annotation prompt, and the document prompt directs the annotator to judge a document's subject rather than its vocabulary.

Document annotation. The annotator is Qwen3-30B-A3B-Instruct-2507, taking a majority vote over five samples. It labels 50,000 randomly sampled FineWeb-Edu documents, which we split by document into training, development and test sets. On 300 held-out documents that we labelled by hand, the annotator reaches an F_1 of 0.89.

C.2 DOCUMENT CLASSIFIER

Document classifier. We train a logistic-regression classifier on frozen `bge-base-en-v1.5` representations. Documents exceed the encoder's context window, so we embed each document in chunks that fill it, L_2-normalise each chunk's `[CLS]` embedding, and average the chunks into one document embedding, which keeps every token of a document inside its representation. We select the classification threshold by maximizing F_1 on the development split and apply it unchanged to the test split, where the classifier achieves an F_1 of 0.78 and an AUROC of 0.98.

C.3 FORGET SET $\mathcal{D}_F$

Selecting the medical (target-domain) documents. $\mathcal{D}_F$ consists of the highest-scoring documents that together account for 5% of corpus tokens, which sets its threshold at 0.75. We set this cut for precision: suppression trains against every document in $\mathcal{D}_F$, so a general document admitted to it becomes a negative example for retained capabilities. On the test split, the threshold reaches a precision of 0.94 at a recall of 0.51.

Selecting the tokens. Token-level suppression applies its loss only to the tokens of $\mathcal{D}_F$ selected by the token mask m. The default mask keeps tokens that the TOKEN FILTERING probe (Appendix D.1) labels medical at its F_1-maximizing threshold. On a representative sample of $\mathcal{D}_F$-like documents, the probe's training labels mark 71% of tokens as medical, and the F_1 mask selects 73%. We therefore use this as the default token-level mask. We additionally evaluate a more conservative *confident* mask that keeps only the 20% of tokens receiving the highest medical scores. We choose 20% as a deliberately restrictive operating point to test whether concentrating suppression on the highest-confidence tokens reduces spillover to retained content.

D EVALUATION

D.1 BASELINES

Filtering. The classifier of Appendix C scores every document, and we remove the highest-scoring documents until they account for $\rho = 20\%$ of corpus tokens. We set this cut for recall: on the test split, it removes 99% of medical documents at a precision of 0.45. At the F_1-maximizing threshold, 15% of medical documents would remain in the pretraining corpus. Medical documents that pass the filter stay available throughout pretraining, and a model's target-domain loss under filtering falls quickly when even a few target-domain documents pass the filter (Shilov et al., 2025). O'Brien et al. (2026) likewise set their filters to minimize false negatives, and Rathi & Radford (2026) find that trading precision for recall makes even weak classifiers effective filters given enough pretraining compute, at a cost to performance across the board.

Token filtering. We follow the token-level filtering of Rathi & Radford (2026). A linear probe on the concatenated layer-13 representations of a left-to-right and a right-to-left 224M language model, both pretrained on FineWeb-Edu, scores every token; end-of-document tokens are never hidden. We train the probe on token labels derived from sparse-autoencoder latents of Gemma 2 9B as in their work, keeping medical latents consistent with the category definition of Appendix C.1 and requiring three active latents per medical token. We parameterize the filtering threshold by the share of corpus tokens hidden, following Rathi & Radford (2026). For Table 1, we select a 15% filtering rate at 113M to match the general loss of suppression and keep this rate fixed across model sizes. To examine the trade-off beyond this matched-loss comparison, at 113M we additionally evaluate 9.2%, 20%, and 50% filtering, where 9.2% is the F_1-maximizing threshold (Figure 7).

Post-hoc baselines. Following Li et al. (2024), we take the loss at the middle layer and update that layer and the two preceding it, targeting MLP layers only. RMU scales the random vector by a coefficient c. Activation norms differ across models and optimizers, so the same c is a different intervention in each; we therefore set $c = \beta\|h\|$, where $\|h\|$ is the mean activation norm at the steered layer, and tune and report β. We give RMU the same medical-data budget as suppression during pretraining: across all RMU updates, the forget batches contain the same total number of FineWeb-Edu tokens from $\mathcal{D}F$ as were processed across all suppression steps during pretraining. Each step additionally draws an equal-sized retain batch from $\mathcal{D}\rho$; the retain term penalizes the squared distance between the middle-layer activations of the updated model and those of the frozen starting model, weighted by α. We select α, β, the learning rate and the steered layers by grid search; Table 10 lists the values searched and used. Among configurations with approximately the same general loss as suppression, we report the one requiring the most PubMed fine-tuning tokens to reach the PubMed loss of NO FILTERING. General loss is measured on held-out FineWeb-Edu documents that pass the $\rho = 20\%$ filter, relative to NO FILTERING. Gradient ascent (Jang et al., 2023), SIMNPO (Fan et al., 2025) and CIRCUIT BREAKERS (Zou et al., 2024), run at matched general loss with `AdamW`, resisted less, so we report RMU alone. Gradient ascent is the post-hoc counterpart of suppression during pretraining: it applies the same loss to the same documents $\mathcal{D}F$, but to the finished model, with a KL term to the model's starting predictions on $\mathcal{D}\rho$ in place of the interleaved pretraining steps. We report it with `AdamW` at 521M in Table 11 for completeness.

Table 9: Post-hoc baselines at 521M. General loss is measured on held-out FineWeb-Edu documents that pass the 20% filter and PubMed loss on the held-out PubMed split, both before fine-tuning and relative to NO FILTERING; the last column is the number of PubMed fine-tuning tokens needed to reach the PubMed loss of NO FILTERING, divided by that of FILTERING.

Method	Optimizer	General loss ↓	PubMed loss	Fine-tuning tokens ÷ FILTERING ↑
FILTERING	`Muon`	-0.02	$+0.23$	$1.00\times$
Gradient ascent	`AdamW`	$+0.07$	$+0.51$	$1.02\times$
DETER	`Muon`	$+0.04$	$+11.15$	$\mathbf{13.53}\times$

Table 10: RMU hyperparameters.

	Searched		Used		
	AdamW	Muon	113M	224M	521M
Retain weight α	0.1, 1, 10, 50, 100, 1000, 1200	25, 50, 100, 125, 150, 175, 250, 500, 1000	125	50	25
Steering strength β	0.5, 2.5, 5, 7.5, 10, 12.5, 25	5, 12.5	5	5	5
Learning rate	$3, 4, 5, 6\times10^{-5}$; $1, 2, 3, 6\times10^{-4}$; $1, 2\times10^{-3}$	4×10^{-5}; $3, 4, 6\times10^{-4}$; $1, 3\times10^{-3}$	1×10^{-3}	4×10^{-4}	3×10^{-4}
Steered activation	MLP, block	MLP	MLP	MLP	MLP

Post-hoc optimizer choice. Our main post-hoc experiments use the optimizer with which each model was pretrained. Since optimizer choice can affect post-hoc unlearning (Zhong et al., 2025), we repeat the post-hoc baselines with AdamW for the 521M model, which is pretrained with Muon. Switching to AdamW does not increase resistance for either baseline (Table 11), indicating that their limited resistance is not explained by the post-hoc optimizer.

Table 11: **Changing the post-hoc optimizer does not increase resistance at 521M.** General loss is measured on held-out FineWeb-Edu documents that pass the 20% filter, and PubMed loss on the held-out PubMed split; both are reported before fine-tuning and relative to NO FILTERING. The final column gives the number of PubMed fine-tuning tokens required to reach the PubMed loss of NO FILTERING, normalized by FILTERING. We average over three fine-tuning seeds. DETER denotes suppression during pretraining.

Method	Optimizer	General loss ↓	PubMed loss	Fine-tuning tokens ÷ FILTERING ↑
FILTERING	Muon	-0.02	$+0.23$	$1.00\times$
Gradient ascent	Muon	$+0.05$	$+0.46$	$1.06\times$
Gradient ascent	AdamW	$+0.07$	$+0.51$	$1.04\times$
RMU	Muon	$+0.05$	$+3.13$	$1.40\times$
RMU	AdamW	$+0.05$	$+3.10$	$1.27\times$
DETER	Muon	$+0.04$	$+11.15$	$\mathbf{13.53}\times$

D.2 CORPUS EVALUATION SETS

For the bioRxiv evaluation, we group eight author-assigned subjects as medical-adjacent: neuroscience, immunology, microbiology, cancer biology, physiology, pharmacology and toxicology, pathology, and epidemiology. The remaining 18 subjects form the biology set.

The PubMed evaluation set is drawn from a Common Pile (Kandpal et al., 2025) shard disjoint from those used for adversarial fine-tuning.

D.3 TASK LOSS

We score every benchmark using the cloze formulation of Rathi & Radford (2026) for their base models: the model reads `Question:`, followed by the question, a newline, and `Answer:`, and we compute the negative log-likelihood of the correct answer text, prefixed by a space. We include no answer choices or demonstrations. We average the negative log-likelihood over the tokens of the answer. All our models share one tokenizer, so we normalize by tokens; Bhagia et al. (2025) instead normalize by bytes to reduce the effect of differing tokenizers. We first average over items within each benchmark, then weight benchmarks equally within each group: medical (MedQuAD, MedMCQA, MedQA-USMLE and MMLU medicine), biology (MMLU biology) and general (MMLU STEM and MMLU non-STEM). MMLU medicine consists of the six subjects used by Rathi & Radford (2026): anatomy, clinical knowledge, college medicine, medical genetics, professional medicine and virology. MMLU biology consists of college and high school biology, MMLU STEM of the remaining subjects in MMLU's STEM category, and MMLU non-STEM of all other subjects except human aging, nutrition and human sexuality, which concern health and are excluded from all three groups. We use the MedMCQA validation set because its test labels are

not public. For the 114 MedQuAD items whose question–answer sequence exceeds the 2,048-token context, we truncate the answer to fit; every other item fits. Table 12 lists the evaluation splits and number of items.

To reduce sensitivity to checkpoint-to-checkpoint variation, we average task loss over four checkpoints from the final 10% of training, following Magnusson et al. (2025). At each checkpoint, we subtract the task loss of NO FILTERING at the corresponding training step and report the mean difference and standard deviation across checkpoints. For RMU, we apply the same edit with the same hyperparameters independently to the corresponding FILTERING checkpoint. Corpus loss and adversarial fine-tuning use the final checkpoint.

Table 12: Benchmarks used for task loss.

Benchmark	Split / subset	Items
Medical		
MedQuAD	Answered questions	16,407
MedMCQA	Validation	4,183
MedQA-USMLE (4 options)	Test	1,273
MMLU medicine (6 subjects)	Test	1,111
Biology		
MMLU biology (2 subjects)	Test	454
General		
MMLU STEM (16 subjects)	Test	2,564
MMLU non-STEM (30 subjects)	Test	9,253

D.4 ADVERSARIAL FINE-TUNING

The adversary. The adversary fine-tunes the final pretraining checkpoint with the optimizer the model was pretrained with. A mismatch between the fine-tuning and pretraining optimizers weakens fine-tuning (Liu et al., 2025; Qu et al., 2026), and against our `Muon` models an `AdamW` adversary is slower. Following Rathi & Radford (2026), gradients are clipped at norm 0.5, each step takes 40,960 tokens as 20 sequences of 2,048, and the moment estimates start fresh. With `Muon`, momentum is 0.95 and all parameter groups share one peak rate. We report the average over three fine-tuning seeds for `Muon` and a single seed for `AdamW`.

Table 13: Selected adversary hyperparameters on PubMed and fine-tuning tokens to parity.

Model	Method	Peak learning rate	Tokens to parity (M)
Pretrained with `Muon`			
113M	FILTERING	2×10^{-3}	0.13
	DETER ($c=5$)	2×10^{-3}	0.39
	RMU	2×10^{-3}	0.17
224M	FILTERING	2×10^{-3}	0.12
	DETER ($c=5$)	2×10^{-3}	0.47
	RMU	2×10^{-3}	0.17
521M	FILTERING	2×10^{-3}	0.12
	DETER ($c=5$)	2×10^{-3}	1.58
	RMU	2×10^{-3}	0.16
Pretrained with `AdamW`			
113M	FILTERING	5×10^{-4}	0.27
	DETER ($c=2$)	10^{-3}	0.53
	RMU	5×10^{-4}	0.32
224M	FILTERING	5×10^{-4}	0.19
	DETER ($c=3$)	10^{-3}	0.64
	RMU	10^{-3}	0.29

Adversarial Fine-tuning on PubMed. The adversary fine-tunes on the PubMed section of the Common Pile (Kandpal et al., 2025) for up to 100 steps, taking one pass without replacement, and we score medical log-perplexity after every step on a held-out PubMed validation split. For the largest model size we train (521M), we continue the attack to 8M tokens (0.08% of pretraining) to test whether the gap to FILTERING persists beyond this point. The target is parity, NO FILTERING's own loss on that split, which every method reaches inside the budget (Figure 10). We use weight decay 0.01. For an adversary with `AdamW`, we search over rates from $\{10^{-5}, 5\times10^{-5}, 10^{-4}, 3\times10^{-4}, 5\times10^{-4}, 10^{-3}, 2\times10^{-3}, 3\times10^{-3}, 5\times10^{-3}\}$ and warmups from $\{0, 3, 6, 10, 40\}$ steps. With `Muon`, we search over rates from $\{3\times10^{-4}, 10^{-3}, 2\times10^{-3}, 5\times10^{-3}, 10^{-2}\}$ and warmups from $\{0, 3, 10\}$ steps. For each model we select the learning rate that reaches parity in the fewest steps (Table 13). In our warmup ablations, a 3-step warmup reaches parity fastest with `AdamW`, and no warmup reaches parity fastest with `Muon`. For every model, the selected rate lies inside the searched range: the next lower and next higher rates both reach parity more slowly or not at all. We report resistance as the ratio of a model's fine-tuning tokens to parity to those of the filtered baseline.

E ADDITIONAL RESULTS WITH MUON

Effect on each evaluation. Figure 11 breaks down the increase in loss from suppression relative to FILTERING for each evaluation set. Across model sizes, the increase is consistently larger on medical evaluations than on general-domain evaluations. Among the biology evaluations, the medically adjacent bioRxiv split is affected most at every size.

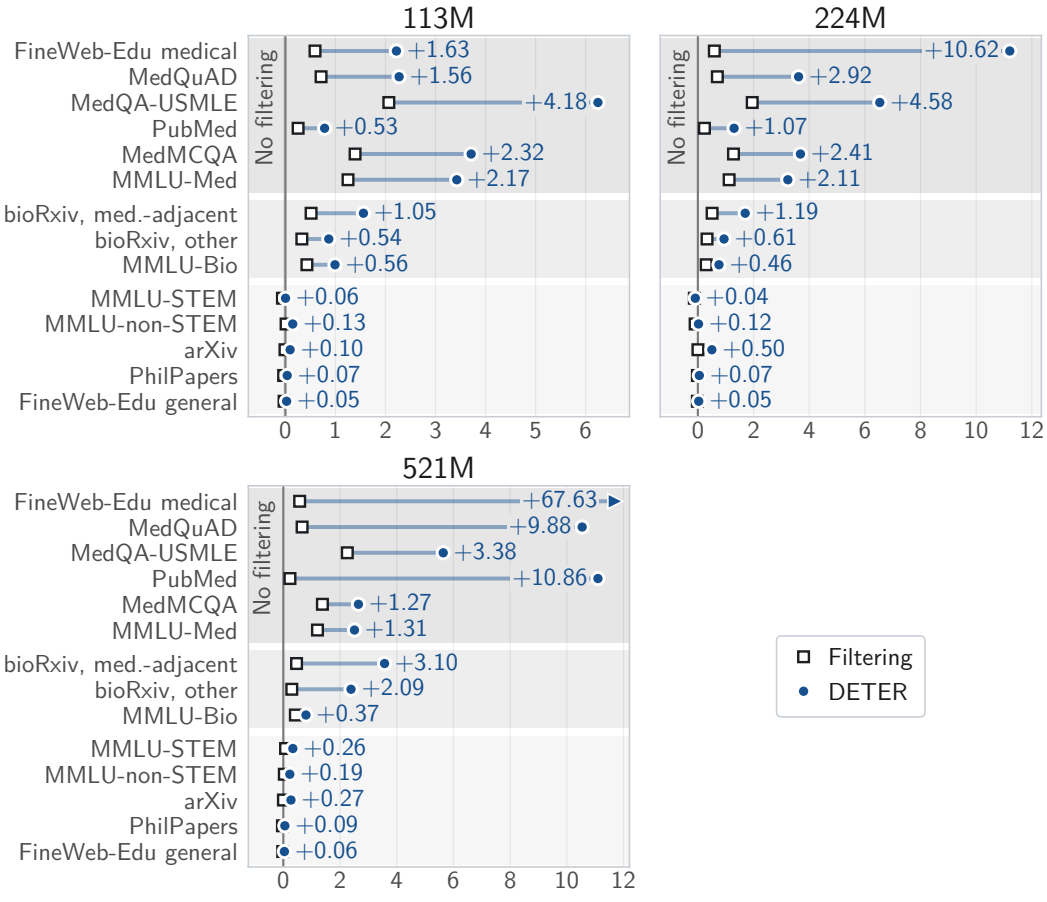


Figure 11: **Suppression primarily affects medical evaluations, with smaller effects on biology and general-domain evaluations.** We report log-perplexity relative to NO FILTERING. Squares show FILTERING and circles show suppression during pretraining; annotations give the increase from FILTERING to suppression for each evaluation.

What remains on downstream tasks after reaching the NO FILTERING PubMed loss? Because our resistance metric is defined by held-out PubMed loss, we also measure downstream task loss when that criterion is first met. For each method, we evaluate the first checkpoint whose PubMed loss matches NO FILTERING (Table 14). These checkpoints occur after different amounts of fine-tuning, so the table is not a matched-compute comparison; Figure 12 shows the task-loss trajectories at matched fine-tuning budgets for the 521M model.

Table 14: **Task losses before fine-tuning and when each method first reaches the PubMed loss of NO FILTERING.** We fine-tune each model on PubMed until its held-out PubMed loss first reaches that of NO FILTERING, and report task loss at the final pretraining checkpoint (before) and at that checkpoint (after). All methods in the right block are therefore evaluated at the same PubMed loss, but reach it after different amounts of fine-tuning, reported as PubMed fine-tuning tokens. **The right block is not a matched-compute comparison:** each method is evaluated at its own checkpoint where PubMed loss first reaches that of NO FILTERING. Figure 12 instead compares task loss after equal amounts of PubMed fine-tuning at 521M. Medical task loss is averaged over MedQuAD, MedMCQA, MedQA-USMLE, and MMLU medicine; general task loss is measured on MMLU STEM and non-STEM; and closest-retain task loss on MMLU biology. All task losses are reported relative to the final checkpoint of NO FILTERING. We average over three fine-tuning seeds. DETER denotes suppression during pretraining.

		Before medical fine-tuning on PubMed			At first checkpoint reaching the PubMed loss of NO FILTERING			
Size	Method	General task loss ↓	Closest-retain task loss ↓	Medical task loss ↑	PubMed FT tokens (M)	General task loss	Closest-retain task loss	Medical task loss
113M	FILTERING	−0.02	+0.43	+1.36	0.13	+0.11	+0.59	+1.39
	TOKEN FILTERING	+0.13	+0.77	+2.40	0.16	+0.27	+0.84	+2.14
	RMU	+0.13	+0.92	+3.58	0.17	+0.15	+0.69	+1.54
	DETER	+0.07	+0.99	+3.91	0.39	+0.40	+1.13	+2.84
224M	FILTERING	−0.11	+0.30	+1.26	0.12	0.00	+0.47	+1.32
	TOKEN FILTERING	+0.02	+0.69	+2.54	0.18	+0.18	+0.83	+2.29
	RMU	−0.03	+0.78	+3.79	0.17	+0.01	+0.50	+1.43
	DETER	−0.03	+0.76	+4.27	0.47	+0.26	+0.87	+2.71
521M	FILTERING	+0.06	+0.42	+1.37	0.12	+0.15	+0.66	+1.54
	TOKEN FILTERING	+0.15	+0.73	+2.64	0.18	+0.31	+0.91	+2.34
	RMU	+0.24	+1.12	+5.35	0.16	+0.16	+0.67	+1.63
	DETER	+0.28	+0.80	+5.33	1.58	+0.71	+1.18	+2.32

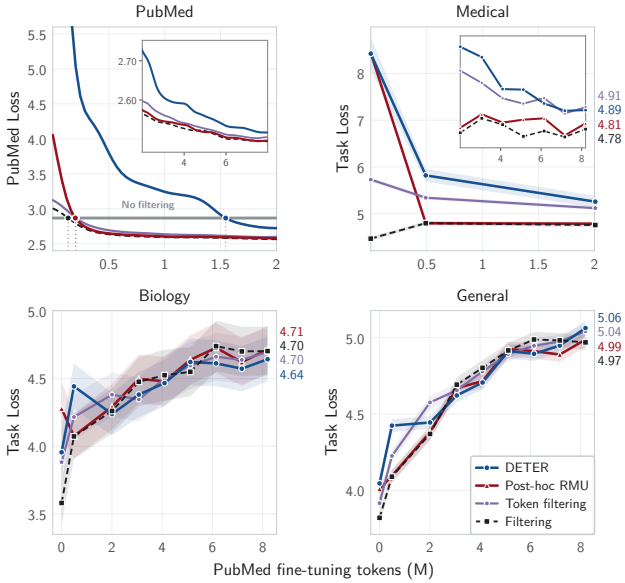


Figure 12: **At 521M, post-hoc RMU returns to the medical task loss of FILTERING early in PubMed fine-tuning, while suppression during pretraining keeps medical task loss higher for longer.** We fine-tune each model on PubMed and report held-out PubMed loss and task loss on the medical benchmarks (MedQuAD, MedMCQA, MedQA-USMLE, and MMLU medicine), MMLU biology, and MMLU STEM and non-STEM (general). The horizontal line marks the PubMed loss of NO FILTERING. Numbers at the right give the task loss at the end of fine-tuning. Results are for one fine-tuning seed; shaded bands are 95% bootstrap intervals over benchmark items. DETER denotes suppression during pretraining.

Suppression interval and start time. Table 2 keeps the number of suppression steps fixed across onsets, so later starts suppress more often. To separate the interval from the onset, we also suppress from the start of pretraining at 113M with the same interval as the 60% start. Although this model takes more suppression steps at approximately the same general loss, it requires less target-domain fine-tuning to reach the NO FILTERING loss (Table 15). The shorter interval therefore does not account for the advantage of starting suppression later.

Table 15: **Using the shorter interval $P = 20$ from the start of pretraining requires less target-domain fine-tuning to undo than starting at 60%.** General loss is measured on held-out retained FineWeb-Edu, relative to NO FILTERING. The final column gives the number of PubMed fine-tuning tokens needed to reach the PubMed loss of NO FILTERING, normalized by the corresponding value for FILTERING at the same model size. We average over three fine-tuning seeds.

	Before fine-tuning	Target-domain fine-tuning
Start	General loss ↓	Fine-tuning tokens ÷ FILTERING ↑
113M, $P = 20$		
0%	$+0.03$	$2.77\times$
60%	$+0.03$	$\mathbf{3.67}\times$