What to Forget in Unlearning? Forget Set Curation for Language Models

Anonymous Authors[1]

Abstract

Machine unlearning aims to remove targeted data or behaviors from a trained model without retraining from scratch. Yet most evaluations assume that the examples to forget are already known. In realistic language-model deployments, a requester may ask a model to stop reproducing a song or book without knowing which spans, documents, quotations, or near-duplicates in a trillion-token corpus support that behavior. We study this missing upstream problem, *forget set curation*: mapping a suppression request to the data passed to an unlearning algorithm. We introduce CLEANSLATE, a benchmark for verbatim output suppression over songs and books, with model-specific extraction profiles, content-grounded QA, and capability-retention evaluations. CLEANSLATE exposes two failure modes. Natural lexical and exact-substring curators often yield forget sets that lead to weak suppression. An evaluation-aware curator suppresses requested continuations almost completely, but causes collateral regression on non-requested content and model-dependent capability loss. These results show that practical unlearning is not only an optimization problem once a forget set is given: the data chosen for forgetting determines both what can be unlearnt and what else is damaged.

1. Introduction

Machine unlearning aims to remove targeted data or behaviors from a trained model without retraining (Ginart et al., 2019; Yao & Xu, 2024), an increasingly relevant goal in privacy, safety, and copyright settings. Yet most unlearning methods and benchmarks study the problem only after the forget set has been supplied (Maini et al., 2024; Shi et al., 2024; Li et al., 2024; Dorna et al., 2025). But a requester may ask a model to stop reproducing a song or book, while the unlearning algorithm requires concrete spans to optimize against. We study this missing upstream step as *forget set curation*, the problem of selecting intervention data from a suppression request.

We focus on verbatim output suppression for culturally embedded works such as songs and books. The desired behavior is selective, with protected continuations becoming difficult to elicit while factual knowledge about the work and unrelated capabilities remain intact. We operationalize this using probabilistic extraction methods that test whether a model assigns high probability to an exact suffix given its prefix (Hayes et al., 2025; Cooper et al., 2025).

Songs and books expose why curation is hard. As shown in Fig 3, lyrics and passages appear across training corpora through copies, quotations, reviews, fan forums, news, code, and incidental discussion. A curator must recover enough of this distributed footprint to suppress the requested behavior without damaging non-requested content or general capabilities. Exact matching alone is insufficient, since models verbatim complete text even when exact n-gram matches are removed from training (Liu et al., 2025b).

Contributions. We introduce CLEANSLATE, a benchmark and evaluation protocol that scores the full pipeline from suppression request to curator to fixed unlearning algorithm to updated model. Our main findings are

1. **Cultural works have diffuse corpus footprints.** Newer works inherit older language through idioms, genre templates, public-domain quotations, and repeated phrases, so the evidence supporting a target continuation may predate the work itself or appear in sources that do not look like copies.
2. **Natural curators are weak and poorly targeted.** BM25 and exact-substring retrieval often fail to produce strong forget-side suppression, and when they do affect the model, the effect is not selective.
3. **Evaluation aware curation reveals a selectivity gap.** Selecting the windows used for evaluation directly suppresses requested continuations almost completely, but also causes substantial model-dependent capability regressions and suppresses non-requested content.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by *The Impact of Memorization on Trustworthy Foundation Models* Workshop @ ICML. Do not distribute.

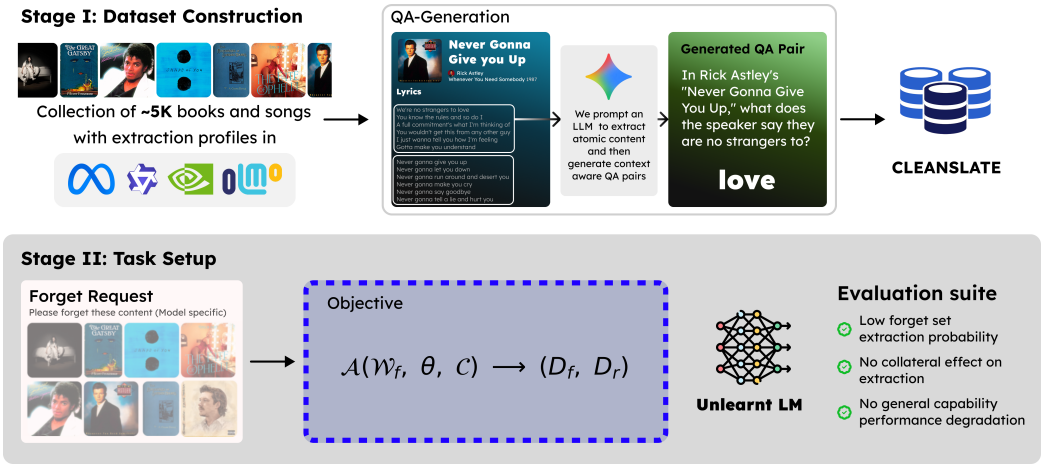

Figure 1. **CleanSlate overview.** *(Top)* We assemble ∼5K books and songs and measure baseline extraction profiles across different model families and generate QA pairs for each content. Together these form the CleanSlate dataset. *(Bottom)* Given a forget request $\mathcal{W}_f$ specifying k items, target model parameters θ, and an arbitrary corpus $\mathcal{C}$, the objective is to create a curator $\mathcal{A}$ that outputs a forget set $\mathcal{D}_f$ and retain set $\mathcal{D}_r$. These are passed to a fixed unlearning procedure, and the resulting model is evaluated.

2. Problem Statement: Forget Set Curation

Let θ be a pretrained model, and let $\mathcal{W}$ be a collection of works, such as songs or books. A suppression request identifies a target subset $\mathcal{W}_f \subset \mathcal{W}$ whose verbatim reproduction should be suppressed. Any possible verbatim reproduction of works in $\mathcal{W}_r = \mathcal{W} \setminus \mathcal{W}_f$ should be preserved.

Extractability. We quantify verbatim reproduction through probabilistic extraction (Hayes et al., 2025; Cooper et al., 2025). A work is divided into a sequence of prefix-suffix window pairs (x, z). For a given prefix x, the model assigns a probability $p_z = p_\theta(z \mid x) = \prod_{t=1}^{|z|} p_\theta(z_t \mid x,\ z_{<t})$. We call a window as *extractable* if $p_z \geq \tau$, we use $\tau = 0.001$ from (Cooper et al., 2025).

Curated forget sets. Given a suppression request $\mathcal{W}_f$, a trained model θ, and access to a large search corpus $\mathcal{C}$, a curator $\mathcal{A}$ returns a forget set D_f and, optionally, a retain set D_r. A downstream unlearning algorithm $\mathcal{U}$ then produces an unlearnt model $\theta' = \mathcal{U}(\theta, D_f, D_r)$. In this paper, the curator $\mathcal{A}$ is the object under evaluation: we compare different choices of $\mathcal{A}$ while holding the downstream unlearning procedure fixed unless otherwise stated.

Search Corpora. For our analysis and experiments we use three distinct scale corpora, $\mathcal{C}_{\text{mid}}$ the `Dolmino` midtraining mix (Olmo et al., 2025), $\mathcal{C}_{\text{pre}}$ a subset of the `Dolma3` pretraining mix (Olmo et al., 2025), and $\mathcal{C}_{\text{CC25}}$ the Jan 2025 Common Crawl snapshot. See App C for details.

Evaluation. The goal of verbatim output suppression is to make extractable windows from $\mathcal{W}_f$ un-extractable after unlearning, without inducing the same effect on extractable windows from $\mathcal{W}_r$. Verbatim suppression should not erase knowledge about a work, we also evaluate content-grounded question answering over the same works, together with general capability benchmarks. Thus, a curator is judged not by textual relevance alone, but by the behavior of the unlearnt model it induces. Figure 1 captures the CLEANSLATE pipeline, see App E for further details.

3. Why is Forget Set Curation Hard

The training evidence supporting a verbatim continuation need not be isolated to a canonical copy of the work. It may appear in lyric aggregators, forum discussions, fan fiction, or code snippets. A curator's true target is therefore a work's *corpus footprint*: the distributed set of documents and spans that can support the target continuation.

Measuring literal overlap. To quantify the corpus footprint of a work, we measure the scale of exact word-level overlaps between the target text and a corpora $\mathcal{C}$ using Infini-gram-mini (Xu et al., 2025). For a given work, we compute localized n-gram occurrence counts across every position (see App D for details). This is a conservative measure as changes in formatting, whitespace, or even case can break a match. It also misses paraphrases, translations, and semantic references, nor does it identify which documents actually caused a continuation. Instead it, lower bounds the literal overlap that a curator would miss if it searched only for canonical copies. At web scale, any large corpus will contain many short n-grams from almost any English text even when no canonical copy is present. So in addition we also measure the *coverage*: the fraction of the requested work covered by N-gram matches.

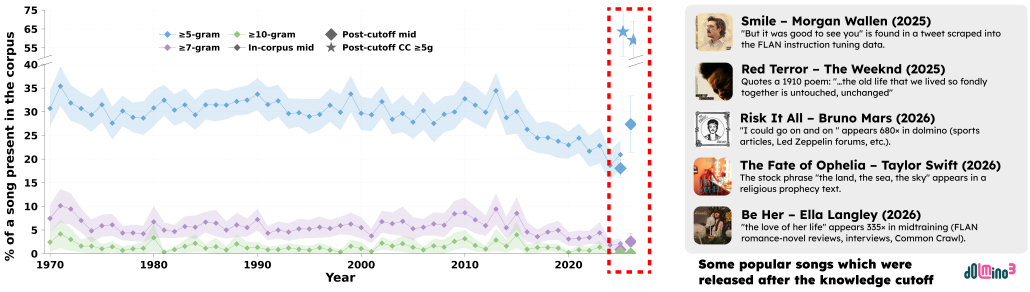

Figure 2. Corpus footprints arise through both outward and inward diffusion. **(Left)** Median per-song coverage in $\mathcal{C}_{\text{mid}}$ across their release years, 1970 to 2026. Older works often have broad footprints from copies, quotations, discussion, and other web sources. These are cases of outward diffusion. **(Right)** Post-cutoff markers show coverage for songs released after cutoff of the corpus. We present five examples, each match older or unrelated sources, including instruction-tuning data, a 1910 poem, forums, religious text, and reviews. These cases illustrate inward diffusion: new works can inherit phrases, quotations, or stock language already present in the corpus.

Outward diffusion. Works with cultural impact spread after release. `Smile` (Morgan Wallen, released 31st Dec 2024) illustrates how rapidly a footprint forms: in $\mathcal{C}_{\text{CC25}}$ (Jan 2025), it exhibits 100% coverage at the 5-gram level, with some spans having $\geq 10^5$ occurrences, it also has 100% coverage at the 50-gram level with some spans having $\geq 10^4$ matches (see Fig 4). $\mathcal{C}_{\text{CC25}}$ contains canonical copies of `Smile`, but also a large cloud of shorter exact overlaps. Older works have had decades to diffuse, they are copied, quoted, discussed, and embedded in unexpected sources. Fig 3 illustrates this for *Never Gonna Give You Up*: beyond canonical sources, the text appears in fan transcriptions, wiki pages, code, reviews, and question-answering sites. Any curator looking solely at canonical sources would miss the vast majority of these spans.

Inward diffusion. A work may contain language that existed in the corpus before the work itself was released. This can happen through idioms, stock phrases, public-domain quotations, or repeated cultural language. Fig 2 shows several post-cutoff songs with substantial exact-overlap coverage in corpora that predate their release. For example, *Red Terror* by The Weeknd has roughly 20% coverage at $n = 7$ in $\mathcal{C}_{\text{mid}}$ because it quotes a 1910 poem. This shows that parts of the requested continuation may be supported by older or unrelated text. This makes curation more subtle than finding noisy and approximate copies of the target work: the relevant evidence may predate the work or appear in documents that do not look like copies at all.

Connection to target continuations. Corpus footprints help explain where verbatim output suppression becomes difficult. Figs 5–8 overlay corpus occurrence counts with extraction probabilities across songs, poems, and books. Across these case studies, extractable continuations often occur in regions with dense literal overlap, such as song choruses, famous quotations, or repeated phrases. While high overlap does not prove that a particular document caused a completion, and low overlap does not rule out elicitation through other mechanisms; the practical implication is that a curator that misses dense regions of the footprint may leave the model enough data to be able to reproduce the requested continuation, while a curator that captures them too broadly may also affect neighboring content.

4. Experiments

We evaluate three curators using CLEANSLATE with $|\mathcal{W}_f| = 50$. Table 1 fixes the unlearner at SimNPO so row differences isolate the curator. Table 5 fixes $\mathcal{A}$ to the evaluation aware curator and varies the unlearner $\mathcal{U}$, (see App H for hyperparameter details). We perform experiments over six models, LLAMA-3.1-8B, OLMO-3-7B, NEMOTRON-9B, QWEN3-8B, GEMMA-3-12B, and OLMO-3-32B. Per-model pre-unlearning evaluations are in App Table 2. We evaluate three retrieval curators (BM25-pre, BM25-mid, Infini-gram-mid) that search large training corpora alongside an evaluation aware (EA) curator that constructs the forget set directly from the reference text, on CLEANSLATE with $|\mathcal{W}_f| = 50$ (full curator details in App I).

4.1. Can retrieval curators induce selective suppression?

We find that retrieval engagement broadly tracks model size (Table 1). The two largest models (GEMMA-3-12B, OLMO-3-32B) show the strongest forget-side movement, LLAMA-3.1-8B and NEMOTRON-9B show partial movement, and the smallest two (OLMO-3-7B, QWEN3-8B) barely move. Retain-side movement follows the same ordering. Selectivity is model-determined rather than retriever-determined. LLAMA-3.1-8B loses between 51.2% and 55.7% of its retain-pool extractability across the three retrieval methods, while its forget side gains at most 28.9%. OLMO-3-32B is the only row where forget movement consistently exceeds retain movement. Switching from corpora, or retrievers, reorders rows but does not change which models engage.

Table 1. Curation comparison at $|\mathcal{W}_f| = 50$ with SimNPO with BM25 pre, BM25 mid, Infini-gram mid, and EA curators. *Forget* ↑ is the fraction of forget-pool extractable windows that fall below the threshold after unlearning; *Retain* ↓ is the same fraction measured on the retain pool, where higher values indicate collateral damage. *QA* Δ is change in CleanSlate-QA accuracy (pp). *Val Avg* is the mean change in validation accuracy (pp) across six benchmarks relative to the per-model baselines in Table 2. The full per-benchmark breakdown appears in the App J.

	Model	Extr. removed (%) F↑	R↓	QA Δ (pp)	Val Avg Δ (pp)
BM25 pre	GEMMA-3-12B	39.3%	38.4%	-2.3	-1.2
	OLMO-3-32B	43.1%	30.8%	-0.2	+1.2
	LLAMA-3.1-8B	28.9%	51.2%	-0.4	+0.3
	NEMOTRON-9B	14.6%	14.9%	+0.6	-2.4
	QWEN3-8B	1.6%	2.1%	-3.9	-0.5
	OLMO-3-7B	1.0%	2.9%	-0.1	-0.5
	Average	21.4%	23.4%	-1.1	-0.5
BM25 mid	GEMMA-3-12B	46.3%	46.0%	-3.2	-3.4
	OLMO-3-32B	43.0%	35.2%	+0.3	+1.4
	LLAMA-3.1-8B	16.7%	54.0%	-1.1	-0.7
	NEMOTRON-9B	13.0%	14.8%	+0.6	-2.6
	QWEN3-8B	2.3%	2.7%	-5.8	-0.6
	OLMO-3-7B	1.5%	3.7%	-0.0	-0.4
	Average	20.5%	26.1%	-1.5	-1.0
Infinigram mid	GEMMA-3-12B	47.8%	44.9%	-2.8	-2.8
	OLMO-3-32B	44.9%	36.0%	+0.4	+1.5
	LLAMA-3.1-8B	19.4%	55.7%	-0.8	-0.2
	NEMOTRON-9B	12.7%	14.5%	+0.5	-0.1
	QWEN3-8B	2.8%	2.8%	-4.6	-0.8
	OLMO-3-7B	1.6%	3.2%	-0.1	-0.2
	Average	21.6%	26.2%	-1.2	-0.4
EA	GEMMA-3-12B	96.2%	85.4%	-6.1	-7.4
	OLMO-3-32B	100.0%	97.0%	-2.7	-0.9
	LLAMA-3.1-8B	100.0%	100.0%	-5.7	-10.4
	NEMOTRON-9B	100.0%	80.3%	-1.2	-9.6
	QWEN3-8B	100.0%	82.7%	-21.9	-3.9
	OLMO-3-7B	100.0%	72.2%	-1.0	-0.9
	Average	99.4%	86.3%	-6.5	-5.5

4.2. What if the target windows are given directly?

EA drives forget extractability to near 0% on every model, including the models that retrieval barely moved (Table 1). The retrieval failure on these models is therefore curator-side rather than a property of the model. Retain extractability also falls on every row. Capability cost varies by an order of magnitude across the suite, with damage concentrated on BBH and LAMBADA.

4.3. Are EA curator's effects algorithm-specific?

To determine if the EA curator's collateral damage is specific to SimNPO, we evaluate two additional unlearning algorithms (UNDIAL and RMU) see Table 5. Across SimNPO and UNDIAL, F reaches 100% on all three models, and R stays between 72–100%. While both algorithms suffer from severe collateral forgetting by degrading retain-pool extractability, they produce different capability outcomes on identical EA inputs. UNDIAL largely preserves average validation accuracy (+0.1 pp) while SimNPO degrades it (−5.1 pp). RMU reduces F on LLAMA-3.1-8B and OLMO-3-7B, with retain extractability falling alongside. The same retain-pool damage and capability gap persist at $|\mathcal{W}_f| = 100$ on LLAMA-3.1-8B and QWEN3-8B (see App L).

5. Discussion and Future Direction

Results suggest that the forget set should not be treated as a fixed premise of language-model unlearning. In real deployments, the request is often stated at the level of a work or behavior, while the unlearning algorithm requires concrete data to update against. What is selected for forgetting determines both whether the requested continuation becomes difficult to elicit and what else is disturbed. Forget set curation is therefore part of the unlearning problem, not merely a preprocessing detail.

Two insufficient endpoints. The experiments expose two incomplete approaches to curation. Off-the-shelf corpus retrieval is weak and poorly targeted: lexical and exact-substring search can recover pieces of a work's corpus footprint, but textual overlap alone does not reliably induce selective verbatim suppression. Conversely, evaluation aware curation removes the retrieval bottleneck by selecting target windows directly, but still causes substantial output suppression of non-requested continuations and model-dependent capability regressions. Thus, the problem is not simply to find more target-like text. It is to construct a forget set whose effects remain localized after the model update.

Toward algorithm-aware curation. This localization depends on the downstream unlearning algorithm. The same evaluation aware forget set produces different capability profiles under SimNPO, UNDIAL, and RMU, suggesting that curation and unlearning should be evaluated jointly rather than as independent modules. Future curators may need to combine corpus-footprint signals with dense or hybrid retrieval, or influence estimation to predict which examples will suppress the requested behavior without unnecessary collateral effects. CLEANSLATE studies a deliberately narrow setting: verbatim output suppression for songs and books. It does not address all forms of unlearning, and our exact-overlap diagnostics miss paraphrase, translation, and semantic reuse. These limitations are also what make the task measurable. Extending forget set curation to richer request types, broader corpora, and algorithm-aware selection is a natural next step. The broader message is that practical unlearning should be studied end-to-end, from request, to curated forget set, to edited model.

References

Ahmed, A., Cooper, A. F., Koyejo, S., and Liang, P. Extracting books from production language models. *arXiv preprint arXiv:2601.02671*, 2026.

Allouah, Y., Guerraoui, R., and Koyejo, S. Distributional machine unlearning via selective data removal. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IPqUBL4R9x.

Cao, P., Wang, C., He, Z., Yuan, H., Li, J., Chen, Y., Liu, K., Zhao, J., et al. Rwku: Benchmarking real-world knowledge unlearning for large language models. *Advances in Neural Information Processing Systems*, 37: 98213–98263, 2024.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., Oprea, A., and Raffel, C. Extracting training data from large language models. 2021. URL https://arxiv.org/abs/2012.07805.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models. 2023. URL https://arxiv.org/abs/2202.07646.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Cooper, A. F., Gokaslan, A., Ahmed, A., Cyphert, A. B., Sa, C. D., Lemley, M. A., Ho, D. E., and Liang, P. Extracting memorized pieces of (copyrighted) books from open-weight language models. 2025. URL https://arxiv.org/abs/2505.12546.

Dong, Y. R., Lin, H., Belkin, M., Huerta, R., and Vulić, I. Undial: Self-distillation with adjusted logits for robust unlearning in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8827–8840, 2025.

Dorna, V., Mekala, A., Zhao, W., McCallum, A., Lipton, Z. C., Kolter, J. Z., and Maini, P. Openunlearning: Accelerating llm unlearning via unified benchmarking of methods and metrics. 2025. URL https://arxiv.org/abs/2506.12618.

Engstrom, L., Feldmann, A., and Madry, A. Dsdm: Model-aware dataset selection with datamodels. *arXiv preprint arXiv:2401.12926*, 2024.

Fan, C., Liu, J., Lin, L., Jia, J., Zhang, R., Mei, S., and Liu, S. Simplicity prevails: Rethinking negative preference optimization for llm unlearning, 2025. URL https://arxiv.org/abs/2410.07163.

Gandikota, R., Feucht, S., Marks, S., and Bau, D. Erasing conceptual knowledge from language models. *arXiv preprint arXiv:2410.02760*, 2024.

Georgiev, K., Rinberg, R., Park, S. M., Garg, S., Ilyas, A., Madry, A., and Neel, S. Attribute-to-delete: Machine unlearning via datamodel matching. *arXiv preprint arXiv:2410.23232*, 2024.

Ginart, A., Guan, M., Valiant, G., and Zou, J. Y. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

Hayes, J., Swanberg, M., Chaudhari, H., Yona, I., Shumailov, I., Nasr, M., Choquette-Choo, C. A., Lee, K., and Cooper, A. F. Measuring memorization in language models via probabilistic extraction. 2025. URL https://arxiv.org/abs/2410.19482.

Ilyas, A., Park, S. M., Engstrom, L., Leclerc, G., and Madry, A. Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*, 2022.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Phan, L., et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Liu, J., Xia, C. S., Wang, Y., and Zhang, L. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation, 2023. URL https://arxiv.org/abs/2305.01210.

Liu, J., Min, S., Zettlemoyer, L., Choi, Y., and Hajishirzi, H. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *arXiv preprint arXiv:2401.17377*, 2024.

Liu, J., Blanton, T., Elazar, Y., Min, S., Chen, Y.-S., Chheda-Kothary, A., Tran, H., Bischoff, B., Marsh, E., Schmitz, M., et al. Olmotrace: Tracing language model outputs back to trillions of training tokens. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 178–188, 2025a.

Liu, K. Z., Choquette-Choo, C. A., Jagielski, M., Kairouz, P., Koyejo, S., Liang, P., and Papernot, N. Language

models may verbatim complete text they were not explicitly trained on, 2025b. URL https://arxiv.org/abs/2503.17514.

Liu, Z., Lin, H., Ran, Y., Zhang, D., Xie, J., Li, C., Zhao, W., and Xu, Z. Randomized antipodal search done right for data pareto improvement of llm unlearning. *arXiv preprint arXiv:2604.16591*, 2026.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. Tofu: A task of fictitious unlearning for llms. 2024. URL https://arxiv.org/abs/2401.06121.

Olmo, T., Ettinger, A., Bertsch, A., Kuehl, B., Graham, D., Heineman, D., Groeneveld, D., Brahman, F., Timbers, F., Ivison, H., et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Pal, S., Wang, C., Diffenderfer, J., Kailkhura, B., and Liu, S. Llm unlearning reveals a stronger-than-expected coreset effect in current benchmarks. *arXiv preprint arXiv:2504.10185*, 2025.

Project Gutenberg. Project gutenberg. https://www.gutenberg.org, 1971. Accessed March 11, 2026.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Reddy, S., Chen, D., and Manning, C. D. Coqa: A conversational question answering challenge, 2019. URL https://arxiv.org/abs/1808.07042.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL https://arxiv.org/abs/1907.10641.

Shi, W., Lee, J., Huang, Y., Malladi, S., Zhao, J., Holtzman, A., Liu, D., Zettlemoyer, L., Smith, N. A., and Zhang, C. Muse: Machine unlearning six-way evaluation for language models. 2024. URL https://arxiv.org/abs/2407.06460.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., and Wei, J. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Wan, Y., Ramakrishna, A., Chang, K.-W., Cevher, V., and Gupta, R. Not every token needs forgetting: Selective unlearning to limit change in utility in large language model unlearning. *arXiv preprint arXiv:2506.00876*, pp. 622–632, 2025.

Wu, R., Yadav, C., Salakhutdinov, R., and Chaudhuri, K. Evaluating deep unlearning in large language models. *arXiv preprint arXiv:2410.15153*, 2024.

Xu, H., Liu, J., Choi, Y., Smith, N. A., and Hajishirzi, H. Infini-gram mini: Exact n-gram search at the internet scale with fm-index. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24955–24980, 2025.

Yao, Y. and Xu, X. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37: 105425–105475, 2024.

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

Zhou, X., Qiang, Y., Zade, S. Z., Zytko, D., Khanduri, P., and Zhu, D. Not all tokens are meant to be forgotten, 2025. URL https://arxiv.org/abs/2506.03142.

Zhu, X., Zhang, M., Liu, O., Jia, R., and Neiswanger, W. Llm unlearning without an expert curated dataset. In *Second Conference on Language Modeling*, 2025.

Appendix

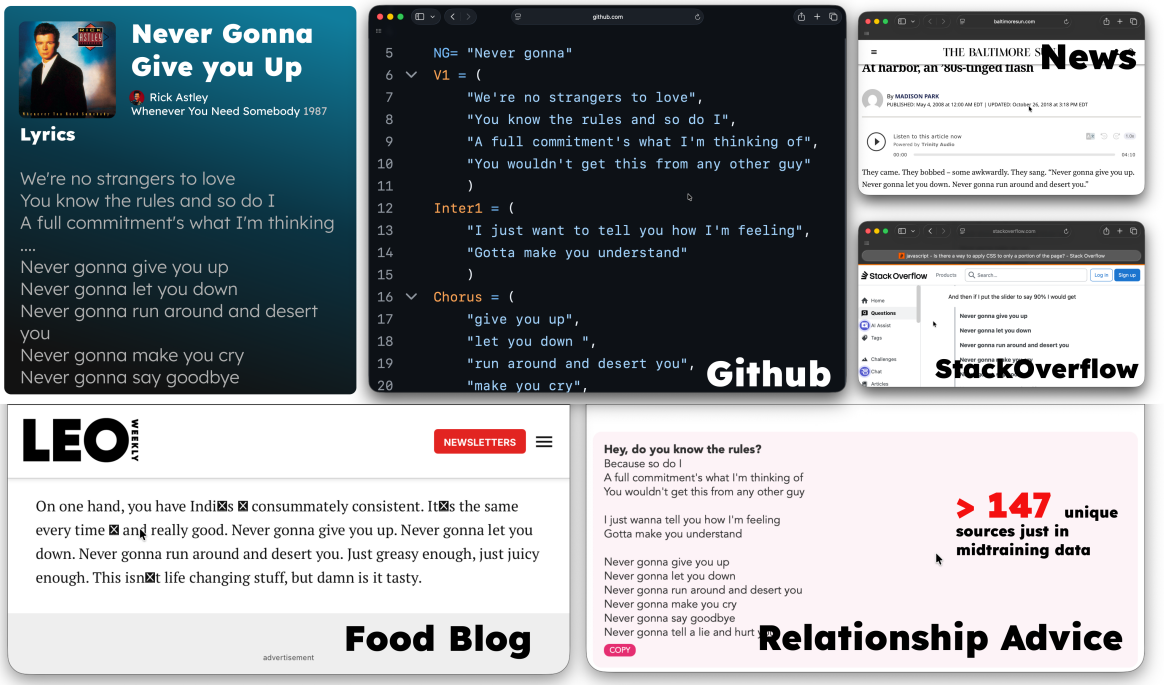

Figure 3. Footprint of songs and books extends far beyond their canonical copy. The top panel shows the canonical source for *Never Gonna Give You Up*, with full lyrics. The surrounding panels show non-canonical sources that echo the same text, including a Python file on GitHub that stores the lyrics, a Baltimore Sun news article that weaves the hook into reporting on a flash mob, a Stack Overflow answer that uses the lines as filler text, a LEO Weekly food review that drops the chorus into prose about a sandwich, and a pickup lines site that recasts the lyrics as relationship advice.

A. Related Work

Memorization and verbatim extraction. Language model memorization is commonly studied through extraction: prompting a model with a prefix and testing whether it reproduces a target suffix (Carlini et al., 2021; 2023; Ahmed et al., 2026). (Hayes et al., 2025) refine this into probabilistic discoverable extraction, measuring whether a target continuation can be produced under repeated sampling. (Cooper et al., 2025) apply this framework to copyrighted books, showing that extractability varies substantially across works and model families. We use this completion-style notion of memorization as a proxy for verbatim output suppression; the goal is not to infer training membership or erase all knowledge of a work, but to make requested continuations difficult to elicit while preserving nearby knowledge and unrelated capabilities.

Machine unlearning algorithms and benchmarks. Machine unlearning aims to remove the influence of specified data or behaviors from a trained model without retraining from scratch (Ginart et al., 2019; Yao & Xu, 2024). A range of algorithms have been proposed for language models. Gradient ascent on the forget loss is the simplest baseline. Negative preference optimization and SimNPO (Zhang et al., 2024; Fan et al., 2025) cast unlearning as preference optimization against a reference model; Representation Misdirection Unlearning (Li et al., 2024) perturbs internal representations on forget data while regularizing retain representations; and UNDIAL (Dong et al., 2025) adjusts logits through self-distillation. Existing benchmarks evaluate whether such updates suppress targeted information while preserving utility, including synthetic biographies in TOFU (Maini et al., 2024), multi-axis evaluation in MUSE (Shi et al., 2024), hazardous-knowledge unlearning in WMDP (Li et al., 2024), and unified evaluation in OpenUnlearning (Dorna et al., 2025). These benchmarks differ in domain and objective, but they typically provide the examples, entities, or evaluation targets to be forgotten. In contrast, we study the upstream curation problem: given a suppression request for a work, what data should be selected for the unlearning update?

Request-level, entity-level, and fact-level unlearning. Some recent work moves closer to settings where the target is specified at a higher level than a fixed forget corpus. RWKU (Cao et al., 2024) studies real-world knowledge unlearning, where the algorithm receives a target entity and the original model rather than an explicit training corpus. It then uses synthetic data produced by the model under evaluation to construct a forget set. This is close in spirit to our setting because the forget set is not directly provided, but the evaluated artifact differs: RWKU primarily measures unlearning algorithms under synthetic forget sets, while we evaluate the curation of forget sets under fixed unlearning algorithms. Deep fact unlearning (Wu et al., 2024) studies whether a target fact remains inferable from retained facts under logical rules. This is related to our setting in that a target behavior can be supported by non-target evidence, but the unit and objective are different: deep unlearning targets factual deductive closure, while we study verbatim continuation and ask which corpus spans or documents should be used for suppression.

Data selection, retrieval, and attribution. Recent work shows that the contents of a forget set matter even after the forget corpus has been specified (Liu et al., 2026). (Pal et al., 2025) find that small subsets of benchmark-provided forget sets can match full-set unlearning, while (Wan et al., 2025) and (Zhou et al., 2025) show that token-level selection within known forget examples can reduce utility loss. (Allouah et al., 2026) formalize distributional unlearning as selective data removal: choosing a small subset whose removal moves the edited data distribution away from the unwanted domain while preserving the retained one. This motivates studying selection itself, but prior work largely assumes that the unwanted examples or target domain are already known. Addressing the upstream dataset construction problem, Zhu et al. (2025) synthesize proxy forget data given only a broad domain name; while related, we focus on measuring the role of the distributed corpus footprint for suppressing verbatim output of specific targeted individual works. A natural approach to this request-level curation is to retrieve text overlapping with the target work, using lexical search or exact-substring systems such as BM25, Infini-gram, and Infini-gram-mini (Liu et al., 2024; Xu et al., 2025). However, textual overlap is only an imperfect proxy for the data responsible for a model continuation: models can verbatim complete text even when exact n-gram matches have been removed from training data (Liu et al., 2025b). Influence functions and datamodeling offer a more causal view of training-example responsibility (Koh & Liang, 2017; Ilyas et al., 2022; Engstrom et al., 2024; Georgiev et al., 2024), but applying them to request-level curation over trillion-token corpora remains an open challenge.

B. Compute Requirements

All experiments were conducted on a compute node equipped with 8 NVIDIA H200 GPUs (141GB VRAM each), 230 CPU cores, 3TB of RAM, and 60TB of local NVMe storage. While model training utilized 1–2 GPUs, evaluation and validation tasks were performed on a single GPU. Notably, the corpus search and forget set curation phases are significantly memory- and storage-bound due to the scale of the datasets involved; these stages necessitated the full utilization of the available system memory and high-speed disk I/O.

C. Search Corpora ($\mathcal{C}$) Details

For our experiments we use three distinct scale corpora

1. $\mathcal{C}_{\text{mid}}$ **(Midtraining):** `allenai/dolma3_dolmino_mix-10B-1025` (Olmo et al., 2025) this is a midtraining mix which has coverage of categories like instruction tuning data (FLAN, Tulu-3-SFT), code (`cranecode`, `stack_edu`), and synthetic reasoning traces (Gemini, QwQ, Llama Nemotron, OpenThoughts).
2. $\mathcal{C}_{\text{pre}}$ **(Pretraining):** A $\sim 11\%$ subset of Dolma-3 6T (`allenai/dolma3_mix-6T`) pretraining mix (Olmo et al., 2025). To make it computationally feasible, we limit this subset to five Common Crawl shards of the full pretraining corpus, `art_and_design`, `entertainment`, `history_and_geography`, `literature`, and `religion`.
3. $\mathcal{C}_{\text{CC25}}$ **(Web Scale):** January 2025 Common Crawl snapshot with $\sim 9T$ tokens, accessed via the index provided by Infini-gram-mini (Xu et al., 2025). Used only for analysis of corpus footprint in Section 3

Olmo et al. (2025) states December 2024 as the knowledge cutoff for $\mathcal{C}_{\text{mid}}$ and $\mathcal{C}_{\text{pre}}$.

D. Computing Coverage of Songs and Books in Large Corpus

We compute these statistics using Infini-gram-mini (Xu et al., 2025) but one can use any other dataset search tool.

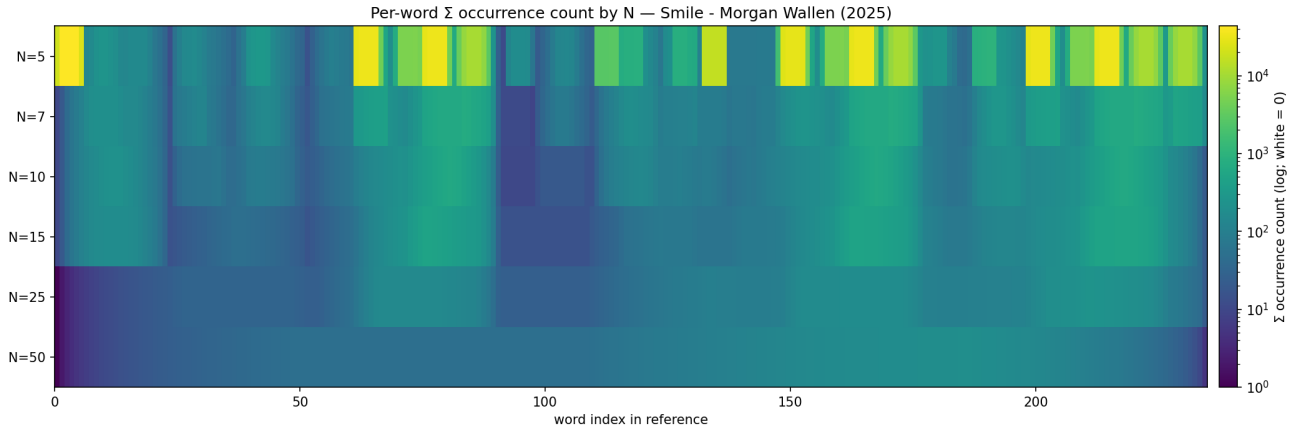

Figure 4. N-gram coverage for *Smile* in $\mathcal{C}_{\text{CC25}}$

D.1. Computing $N-$Gram Matches and Coverage

To quantify the verbatim overlap between a reference sequence and a large-scale corpus $\mathcal{C}$, we employ an iterative retrieval algorithm that identifies the maximal exact n-gram matches starting at every word start position. Let $T_w = (x_1, \ldots, x_m)$ be the whitespace-delimited word sequence of work w. For start position i and length n, let $s_{i,n}$ be the exact character span covering $x_i, \ldots, x_{i+n-1}$, preserving punctuation and spacing. For indexed corpus $\mathcal{C}$, we define the frequency $C_{\mathcal{C}}(s)$ and the coverage statistic $\text{cov}_{\geq N}(w; \mathcal{C})$ as follows:

$$\begin{aligned} C_{\mathcal{C}}(s) &= \sum_{d \in \mathcal{C}} \#_d(s) \\ \text{cov}_{\geq N}(w; \mathcal{C}) &= \frac{1}{m} \sum_{j=1}^{m} \mathbf{1}[\exists\, i, n :\ n \geq N,\ i \leq j < i + n,\ C_{\mathcal{C}}(s_{i,n}) > 0] \end{aligned} \tag{1}$$

The statistic $\text{cov}_{\geq N}$ measures the fraction of word positions in w covered by at least one verbatim n-gram of length $n \geq N$ present in $\mathcal{C}$. For each start position i, we begin at $n_{\min} = 5$ and extend n while $C_{\mathcal{C}}(s_{i,n}) > 0$. Short n-grams ($n < 5$) occur with high background frequency due to linguistic coincidence.

D.2. Aggregate Coverage Statistics

Out of 4,663 works in CLEANSLATE , 4,596 (98.6%) retrieve at least one positive-count 5-gram match with documents in $\mathcal{C}_{\text{mid}}$. The distribution of retrieved documents per work exhibits a median of 119, a 90th percentile of 299, and a maximum of 45,338. For source inspection in $\mathcal{C}_{\text{mid}}$ we sample $D = 20$ documents for each of the top-K ($K = 20000$) spans ranked by length and occurrence count, for $\mathcal{C}_{\text{CC25}}$ we sample $D = 2$ documents for each of the top $K = 2000$ spans, due to computational limitations.

D.3. Extractability vs. Footprint Density Plots

To understand the correlation between localized corpus prevalence and model extractability, we plot the *footprint density* at a given word position j, alongside the maximum extraction probability (p_z) measured amongst all suffixes z covering the position j. The density aggregates the occurrence counts of all valid n-grams in $\mathcal{C}_{\text{mid}}$ that overlap the j-th word of the work.

Figures 5 through 8 corroborate that peaks in $\mathcal{C}_{\text{mid}}$ occurrences strongly align with spikes in extractability. For instance, the choruses of popular songs (*Never Gonna Give You Up*, *Rocket Man*) and famous refrains in poems (*A Dream Within a Dream*) exhibit massive frequency spikes in the corpus. The model's extraction probability neatly mirrors these spikes, rising precisely where the footprint density is highest.

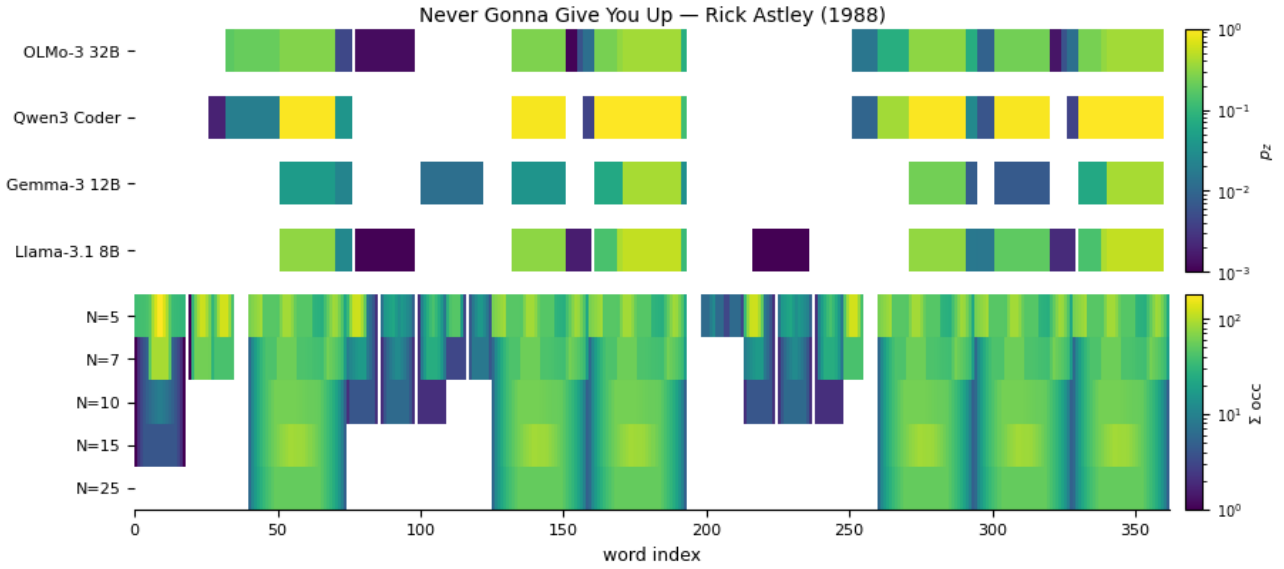

Figure 5. Footprint density (sum of matches of all overlapping n-gram matches in $\mathcal{C}_{\text{mid}}$) vs. maximum extraction probability (p_z) per word for *Never Gonna Give You Up*.

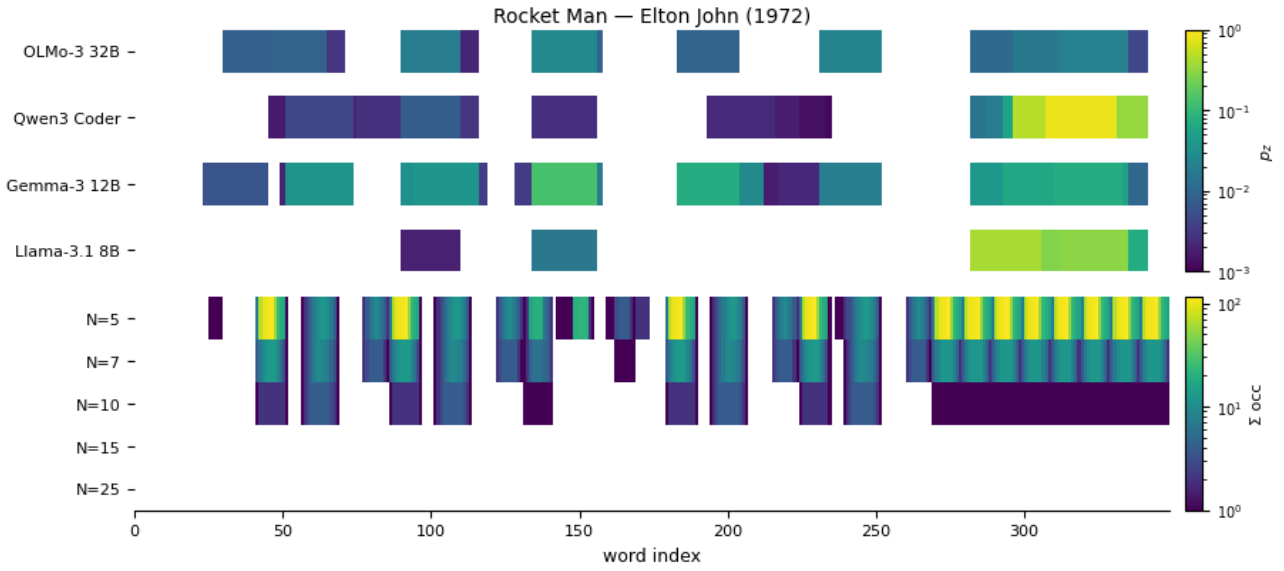

Figure 6. Footprint density vs. maximum extraction probability (p_z) per position for *Rocket Man*.

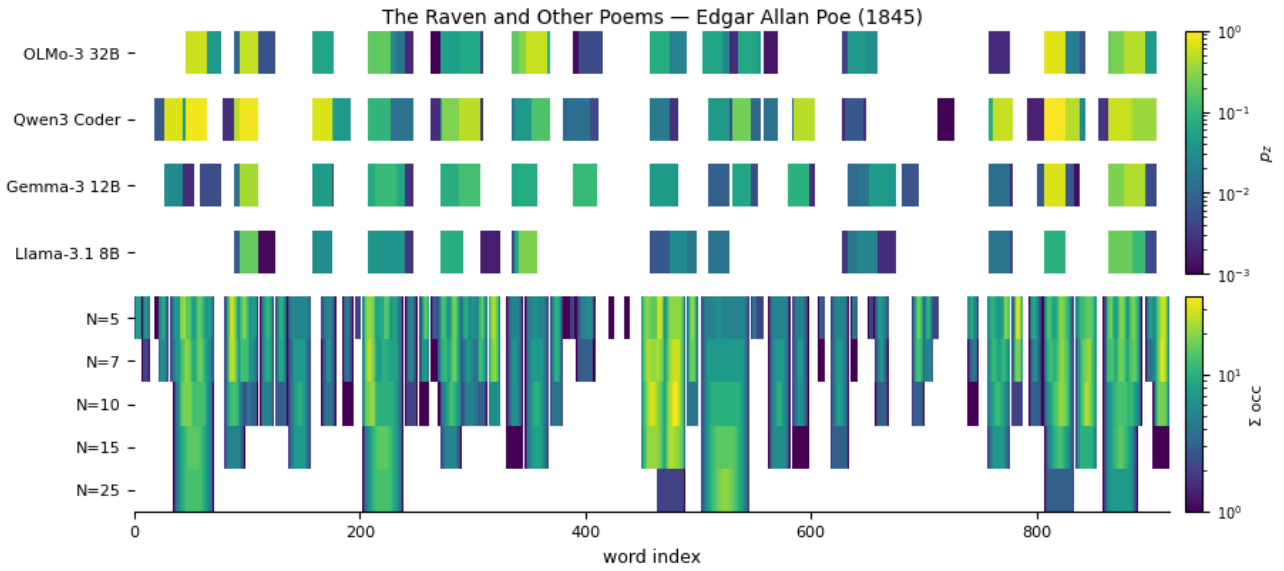

Figure 7. Footprint density in $\mathcal{C}_{\text{mid}}$ vs. maximum extraction probability (p_z) per position for Poesms of Edgar Allen Poe

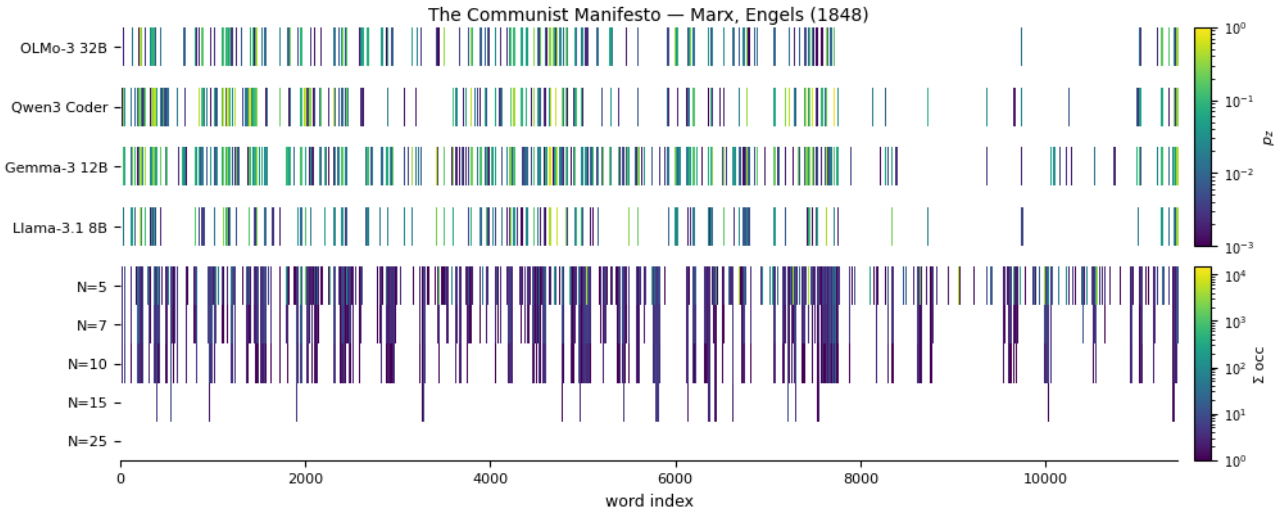

Figure 8. Footprint density in $\mathcal{C}_{\text{mid}}$ vs. maxp_z per position for *The Communist Manifesto*.

E. CLEANSLATE

Building on the formalism of Section 2, CLEANSLATE pairs each suppression request with per-model extractability evidence and content-grounded QA, so that the full pipeline of curator, unlearning algorithm, and edited model can be scored along several axes.

Content domains. CLEANSLATE covers two domains, songs and books, whose corpus footprints (Section 3) are shaped differently. Songs travel as short, repetitive, discrete units. Books propagate as longer-form passages reaching the corpus primarily through commentary and excerption rather than full copies. A curator that recovers one shape may miss most of the other, making the two domains complementary stress tests. We source songs from Billboard Hot 100 annual charts spanning 1970–2025, matched against the LRCLib lyrics database by title and artist, and 50 books, mostly from Project Gutenberg (Project Gutenberg, 1971) with a small number of closed-license works.

Extraction profiles. We slide a window of 100 prefix and 100 suffix characters with stride 10 over each work, and label a work *model-extractable* for a base model θ if at least 5% of its windows are extractable in the sense of Section 2. The resulting profile is per-model. The same song may be extractable for one model and not for another. See App F for more details.

Forget and retain pools. A suppression request is a model specific sample of size $|\mathcal{W}_f| = 50$ drawn from the model-extractable works of θ, with $\mathcal{W}_r$ defined as in Section 2. For a fixed model we reuse the same $\mathcal{W}_f$ and $\mathcal{W}_r$ across curators so that differences are attributable to the curation.

Content-grounded QA. Suppressing verbatim reproduction and erasing factual knowledge are distinct goals. A model that has been asked to stop reproducing a song's lyrics should still be able to answer factual questions about its content. We construct CleanSlate-QA in two stages. First, we prompt an LLM[1] to extract *atomic statements* from each work; factual statements anchored to a named entity, place, number, or concrete event mentioned inside the text, with priors such as title, creator, year, and genre explicitly excluded. Second, each statement is turned into a QA pair whose question embeds the title and creator naturally so it is self-contained, and whose answer is a short 1-5 word entity. For example, in J.K. Rowling's *Harry Potter and the Sorcerer's Stone* we ask where the Dursleys make Harry sleep, with answer *cupboard under the stairs*. The final dataset contains 12,088 QA pairs spanning the songs and books in $\mathcal{W}$.

End-to-end evaluation. For each curator $\mathcal{A}$, we measure baseline extractability and QA on θ, run $\mathcal{A}$ to obtain D_f, apply a fixed unlearning algorithm $\mathcal{U}$ to obtain θ', and re-measure. *Forget* $\uparrow$ is the fraction of extractable windows in $\mathcal{W}_f$ that fall below τ on θ', with higher values better. *Retain* $\downarrow$ is the same fraction over extractable windows in $\mathcal{W}_r$, where high values indicate collateral damage. *QA* Δ is the change in CleanSlate-QA accuracy. A validation suite covers math (GSM8K (Cobbe et al., 2021)), held-out reasoning (BBH (Suzgun et al., 2022)), commonsense (WinoGrande (Sakaguchi et al., 2019)), reading comprehension (CoQA (Reddy et al., 2019)), code (HumanEval+ (Liu et al., 2023)), and language modeling (LAMBADA (Radford et al., 2019)). Per-model baselines are reported in App F.

F. Baseline Extractability Patterns Across Model Families

Extractability is neither uniform across target works nor consistent across model architectures. Table 2 documents the baseline characteristics of our evaluation suite prior to any curation or unlearning. The extraction threshold ($p_z \geq 0.001$) is evaluated over sliding windows of 100 prefix and 100 suffix characters.

Model scale and capability. Larger models consistently exhibit higher raw extractability. For instance, the 32B-parameter OLMO-3-32B achieves a forget-pool extraction rate (Ext-F) of 8.27%, while its 7B-parameter counterpart (OLMO-3-7B) achieves 7.20%. Base models also tend to exhibit higher extractability than their instruction-tuned variants, probably due to alignment training penalizing raw regurgitation in favor of conversational formatting. Repetitive structure of certain songs also increase their extractability, *Drive* by The Weeknd (2025) post dates Llama-3.1-8B and Olmo-32b but both model assign high p_z where a sections of the work are near identical prefix–suffix pair (see Figure 9), this suggests that the language-modeling objective by itself can increase the probability of extraction for highly repetitive works.

[1] We use `gemini-3.1-flash-lite-preview` via LiteLLM.

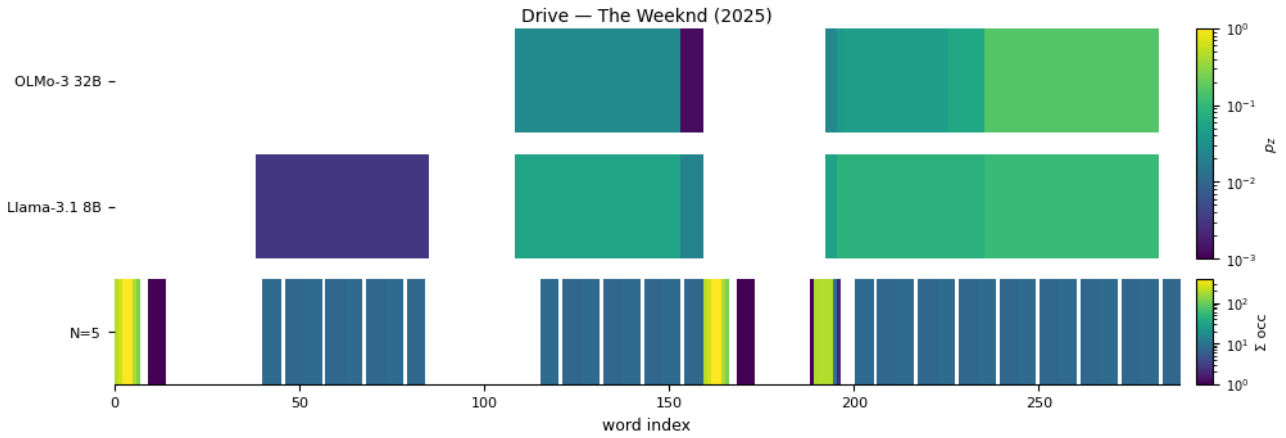

Figure 9. $\max p_z$ across *Drive* by Weekend, both models pre date the song, windows with high p_z are highly repetitive portions of the chorus

Table 2. Baseline evaluation models prior to any curation or unlearning, ordered by forget-pool extractability. The *Forget pool* and *Retain pool* blocks report the total number of $(100, 100)$ character windows (N) and the number that meet the extraction threshold $p_z \geq 0.001$ (*Ext.*). *QA* is CleanSlate-QA pass@5 accuracy (%) averaged over forget and retain items. Remaining columns report baseline accuracy (%) on the validation suite. Forget pool size is $|\mathcal{F}|$=50 except for Olmo-3-7B, which has $|\mathcal{F}|$=38. GSM8K uses flexible-extract scoring.

	Forget pool		Retain pool		QA (%)	Validation accuracy (%)						
Model	N	Ext.	N	Ext.		GSM8K	BBH	HSwag	WinoG	CoQA	HEval+	LMBDA
Llama-3.1-8B	8,513	599	3,999,813	11,130	25.63	50.64	64.57	60.69	74.43	49.10	31.71	76.29
Olmo-3-7B	8,084	582	4,000,242	2,108	27.31	69.60	61.63	54.13	69.77	57.63	31.71	69.63
Nemotron-9B	8,840	647	3,999,486	3,082	22.01	76.80	72.67	58.11	73.48	53.92	10.98	68.00
Qwen3-8B	10,128	755	3,998,198	3,297	34.71	88.55	79.48	57.12	67.88	64.78	58.54	65.17
Gemma-3-12B	27,936	2,210	3,980,390	14,501	30.63	71.80	73.91	62.04	75.30	50.03	12.80	76.03
Olmo-3-32B	67,323	5,565	3,941,003	17,280	35.07	78.24	78.73	60.97	76.48	58.38	39.63	76.31

Heterogeneity of extractable text. The data reveals that extractability is highly localized within the texts themselves. For songs, verses often fall below the extraction threshold, while choruses reinforced by both their internal repetition and their higher external footprint density cross the threshold easily. More importantly, *different models extract different windows*. A specific verse that is highly extractable for LLAMA-3.1-8B may fall below the threshold for NEMOTRON-9B due to differences in their respective training mixtures.

QA performance vs. verbatim extraction. Table 2 also reports the baseline CLEANSLATE-QA accuracy (ranging from 22.01% to 35.07% pass@5). The fact that these models can reliably answer factual questions about the content confirms that they possess abstract knowledge of the works. The core challenge of forget set curation is to selectively suppress the localized p_z spikes responsible for exact extraction while leaving this broader semantic knowledge (QA) and general capabilities (GSM8K, etc.) undisturbed.

G. CleanSlate-QA Benchmark Construction

Here we go over the pipeline used to build CleanSlate-QA, the content-grounded retain-metric benchmark referenced in Section E and reported as the *QA* column of Table 2. Construction has three stages. Stage 1 extracts factual propositions from each work, stage 2 turns each proposition into an atomic question-answer pair, and stage 3 filters candidates by out-of-sample (OOS) model knowledge.

Routing and infrastructure. Each work in $\mathcal{W}$ is routed by length. Books are chunked into 60,000-character windows that are extracted while songs are extracted in a single call. Stages 1 and 2 dispatch `gemini-3.1-flash-lite-preview`. Stage 3 uses (`Qwen/Qwen3.5-9B`, `meta-llama/Meta-Llama-3-8B-Instruct-Lite`).

G.1. Stage 1: Proposition Extraction

The model is asked to extract a small set of *interior factual propositions*, defined as statements anchored to a named entity, place, number, or concrete event mentioned inside the text, with priors such as title, creator, year, and genre explicitly excluded. Each proposition must cite a short source span from the work. The song and book prompts are reproduced below.

Song Proposition Extraction Prompt

You are extracting interior facts from a song's lyrics that will be used to build a retain-metric QA benchmark.

Title: "`{title}`"
Creator: `{creator}`

LYRICS:
`{text}`

TASK: Extract between 3 and 8 **interior factual propositions** from these lyrics.

A good proposition:

- Refers to a named entity, specific detail, number, place, person, action, or relationship mentioned INSIDE the lyrics.
- Is a full factual statement, not just a word.
- Cites a short source span (1–2 lines from the lyrics) where the fact appears.

A bad proposition:

- Is about the title, creator/artist, year, genre (priors).
- Restates the hook, chorus, or title in different words (e.g. for "Ladies' Night": "the song says it is your night" — that's just the hook).
- Has no named entity, place, number, or specific concrete detail.
- Is a generic feeling, mood, theme, or exhortation ("the song tells listeners to dance").
- Uses pronouns/possessives as its key content ("your", "their", "this").

If the song is abstract with few concrete details, return fewer propositions. Do NOT pad.

Return ONLY a JSON object:
`{"propositions": [{"fact": "...", "source_span": "..."}, ...]}`

Book Proposition Extraction Prompt (per chunk)

You are extracting interior facts from a CHUNK of a book that will be used to build a retain-metric QA benchmark.

Book: "`{title}`"
Author: `{creator}`
Chunk `{chunk_idx}` of `{total_chunks}`

TEXT:
`{text}`

TASK: Extract between 5 and 15 **interior factual propositions** from this chunk.

Prioritize:

- Named characters and their distinguishing features, relationships, actions.
- Specific locations, settings, objects mentioned.
- Concrete plot events in this chunk.
- Numeric specifics (ages, dates within the narrative, counts).

Avoid:

- Anything about the title, author, publication year, genre (priors).

- Generic theme/mood statements without a concrete anchor.
- Verbatim famous quotes unless they reveal a concrete fact.

Each proposition must cite a short source span (1–3 lines) from the chunk.

Return ONLY a JSON object:
`{"propositions": [{"fact": "...", "source_span": "..."}, ...]}`

G.2. Stage 2: Atomic QA Generation

Each proposition is converted into one atomic question-answer pair. Questions must be self-contained (naming both title and creator), target an interior detail rather than a prior, and admit a short 1–5 word answer that is itself a named entity, number, specific place, specific object, or proper noun. The model is allowed to return an empty list when no valid pair can be produced for a given proposition. This is the primary mechanism by which low-yield propositions are dropped.

Atomic QA Generation Prompt

Turn this fact into ONE atomic question–answer pair for a retain-metric benchmark.

Title: "`{title}`"
Creator: `{creator}`
Fact: `{fact}`
Source span: `{source_span}`

REQUIREMENTS — the question must:

1. Be SELF-CONTAINED (atomic): name the title AND creator naturally. The model sees only the question, no context.
2. Have a SHORT semantic answer (1–5 words) that is a **named entity, number, specific place, specific object, or proper noun**. Not a pronoun, possessive, or generic colloquial phrase.
3. Target an interior detail — NOT the title, creator, year, or genre.
4. The answer must NOT be the title or any substring of the title/creator.
5. The answer must NOT appear as a phrase inside your own question text.
6. Not be answerable from the title alone (e.g., if the song is "Sexy + 17" don't ask the girl's age).

GOOD examples:

- Q: "In Dr. Hook's 'Sylvia's Mother', how does the caller address Sylvia's mother?" A: "Mrs. Avery"
- Q: "In 'C'mon N' Ride It' by Quad City DJ's, what car does the narrator want to be in the back of?" A: "Impala"

BAD examples:

- Q: "Who sang 'Hotline Bling'?" → priors
- Q: "Complete: 'Ride that choo-choo ___' " → verbatim
- Q: "What is the mood of 'More Than A Woman'?" → subjective
- Q: "In Kool & the Gang's 'Ladies' Night', what does the song say is happening tonight?" A: "your night" → hook-echo, pronoun answer, no specific content

If the fact cannot produce a good atomic Q meeting all requirements, return an empty list.

Return ONLY a JSON object shaped like:
`{"pairs": [{"question": "...", "answer": "..."}, ...]}`

Table 3. Unlearning hyperparameters.

	SimNPO	UNDIAL	RMU
β / δ	2.0 / 0.0	–	–
β_{U}	–	10.0	–
RMU coeff. / layer	–	–	2.0 / `layers.7`
λ_f / λ_r	0.125 / 1.0	1.0 / 1.0	1.0 / 1.0
Retain loss	NLL	NLL	embed-diff
Epochs	1	2	2
LR / warmup	10^{-5} / 10	10^{-5} / 20	10^{-5} / 20
Batch / accum.	8 / 4	8 / 4	8 / 4

G.3. Stage 3: OOS Knowledge Filter and Judge

To drop questions whose answers can only be recovered by memorizing the source work, each candidate question is presented to the OOS probe models with no surrounding context. A candidate is kept iff at least one probe model produces a correct response.

LLM Judge Prompt

Does the model answer contain the reference answer or convey the same meaning?

Reference: `{reference}`
Model Answer: `{model_answer}`

Reply with ONLY this JSON, nothing else: {`"correct": true`} or {`"correct": false`}

H. Unlearning Training Details

The trainer consumes curated forget rows of the form (x, z), where x is a prefix and z is the suffix to suppress. Each training item is anchored on one forget example and paired with a randomly sampled retain example. The total loss is

$$\lambda_f \mathcal{L}_{\text{forget}} + \lambda_r \mathcal{L}_{\text{retain}}.$$

For SimNPO we use the average suffix NLL form

$$\mathcal{L}_{\text{SimNPO}} = -\frac{2}{\beta} \log \sigma\big(\beta(\bar{\ell}_\theta(z \mid x) - \delta)\big)\,,$$

with $\mathcal{L}_{\text{retain}}$ equal to retain NLL. UNDIAL distills from a frozen reference model after subtracting β_{U} from the gold-token teacher logit on forget suffix tokens. RMU minimizes MSE between forget activations and a random control vector at decoder block `model.layers.7`, and uses an activation matching retain loss against the frozen reference model. Table 3 captures all hyperparameters specific to $\mathcal{U}$.

H.1. Some example QA pairs

See Figure 10

I. Retrieval Curators and Forget-Set Construction Details

Curators. To evaluate whether standard corpus search can construct selective forget sets, we test three retrieval curators against large training corpora alongside an evaluation aware baseline. **BM25-pre** and **BM25-mid** use BM25 indices over chunked documents from $\mathcal{C}_{\text{pre}}$ and $\mathcal{C}_{\text{mid}}$ respectively, and for each requested work the curator keeps the top ranked retrieval units and converts them into completion examples. **Infini-gram-mid** uses an Infini-gram-mini index over $\mathcal{C}_{\text{mid}}$ to find maximal exact character spans of the requested text that occur in the corpus, and the matched spans are localized in retrieved corpus contexts. The **evaluation aware (EA)** curator bypasses retrieval and for each work in $\mathcal{W}_f$ constructs D_f from all windows of the reference text. Sequence length is fixed across curators so differences in unlearning behavior reflect forget-example content rather than length.

Figure 10. Example QA pairs form CLEANSLATE

Each retrieval curator maps a request text to a set of prefix–suffix training examples in three steps. First, retrieve corpus units. Second, project retrieved text into evaluation-shaped windows. Third, allocate a bounded number of windows per requested work. The unlearning trainer consumes only these pre-sliced rows, tokenizing the concatenated prefix and suffix and masking prefix tokens from the forget loss.

BM25 retrieval. For BM25, we build sharded BM25s indices over the search corpus. Documents shorter than 100 characters are discarded. Remaining documents are split at word boundaries into segments of at most 2,000 characters with 500-character overlap, and segments shorter than 200 characters are removed. BM25 tokenization lowercases text and removes English stopwords. At search time, each shard returns its top candidates and we retain the global top 100 segments for the requested work. Very long request texts are queried through sampled probes, after which the same global top-k merge is applied.

Infini-gram retrieval. We use an Infini-gram-mini over the corpus. For each character offset i in the request text T, we first test whether the 20-character substring beginning at i occurs in the corpus. If it does, similar to OlmoTrace (Liu et al., 2025a) we binary search for the longest substring $T[i : j]$ with positive corpus count and record the span together with its count. After scanning all offsets, we rank the spans by character length and keep the top 100. For each retained span, the retrieval pipeline queries corpus contexts containing the span. The curation artifact stores the first retrieved context, while metadata records the matched `span_text`, corpus count, coverage, and number of retrieved contexts.

Window extraction. We use the same character window shape for all forget sets. Let $W = 100$ and $S = 10$. A window beginning at character position s emits

$$\text{prefix} = D[s : s + W], \qquad \text{suffix} = D[s + W : s + 2W].$$

EA applies this sliding window directly to the reference text. BM25 applies it to the retrieved BM25 text unit recorded in the retrieval artifact. Infini-gram first locates `span_text` inside the retrieved corpus context and emits stride-spaced windows whose 200-character frame contains the midpoint of the matched span. Duplicate prefix-suffix pairs are removed within each requested work.

Quota allocation. Retrieval can produce many overlapping windows from a small number of documents. To avoid letting a single repeated source dominate the forget set, we cap each requested work at $C = 128$ windows and allocate this budget across retrieved documents. Let v_i be the number of available windows for document i, ordered by retrieval rank. We first give each document up to $F = 4$ windows, bounded by availability and the remaining budget. Any remaining budget is then distributed round-robin over documents in rank order until the cap is reached or no document has unused windows. If a document has more available windows than its quota, we take evenly spaced windows from that document.

Retain data. Following prior work (Gandikota et al., 2024; Li et al., 2024) we use `WikiText` as the retain set D_r, and we keep this fixed across curators to isolate the effect of the forget set D_f. During training, each forget example is paired with a randomly sampled retain example when the unlearning objective uses retain regularization.

J. Curation Method Ablation

See Table 4

K. Unlearning Algorithm Ablation

See Table 5

L. Scaling Forget Request

See Table 6

Table 4. Curation comparison at $|\mathcal{W}_f| = 50$ with SimNPO unlearning, for three retrieval-based curators (BM25 pre, BM25 mid, Infini-gram mid) and the evaluation-aware curator (EA). *Forget*↑ is the fraction of forget-pool extractable windows that fall below the threshold after unlearning; higher is better. *Retain*↓ is the same fraction measured on the retain pool, where higher values indicate collateral damage. *QA* Δ is the change in CleanSlate-QA accuracy (pp). Remaining columns report change in validation accuracy (pp) relative to the per-model baselines in Table 2. Model order is fixed across blocks. *Avg* is the mean of the six validation Δs in the same row.

		Extr. removed (%)		QA	Validation Δ (pp)						
	Model	F↑	R↓	Δ (pp)	GSM8K	BBH	WinoG	CoQA	HEval+	LMBDA	*Avg*
BM25 pre	GEMMA-3-12B	39.3%	38.4%	-2.3	+3.9	-4.7	+0.2	-6.4	+4.9	-4.9	-1.2
	OLMO-3-32B	43.1%	30.8%	-0.2	+1.5	+0.2	+0.6	+4.6	+0.0	+0.5	+1.2
	LLAMA-3.1-8B	28.9%	51.2%	-0.4	-0.1	+0.1	-0.6	+1.9	+1.2	-1.0	+0.3
	NEMOTRON-9B	14.6%	14.9%	+0.6	-11.1	-4.0	+0.9	+0.6	+0.6	-1.7	-2.4
	QWEN3-8B	1.6%	2.1%	-3.9	-1.4	+1.4	+3.2	-3.5	-6.7	+3.8	-0.5
	OLMO-3-7B	1.0%	2.9%	-0.1	+1.7	-0.1	-0.1	-3.8	-0.6	-0.0	-0.5
	Average	21.4%	23.4%	-1.1	-0.9	-1.2	+0.7	-1.1	-0.1	-0.5	-0.5
BM25 mid	GEMMA-3-12B	46.3%	46.0%	-3.2	+2.0	-8.6	-0.4	-9.2	+2.4	-6.3	-3.4
	OLMO-3-32B	43.0%	35.2%	+0.3	+2.4	+0.3	+1.4	+2.9	+0.6	+0.7	+1.4
	LLAMA-3.1-8B	16.7%	54.0%	-1.1	-0.2	-0.4	-1.5	-2.4	+1.8	-1.4	-0.7
	NEMOTRON-9B	13.0%	14.8%	+0.6	-10.1	-4.1	+0.6	-0.3	+0.6	-2.4	-2.6
	QWEN3-8B	2.3%	2.7%	-5.8	-1.1	+1.4	+3.6	-0.6	-10.4	+3.6	-0.6
	OLMO-3-7B	1.5%	3.7%	-0.0	+1.0	-0.2	+0.0	-1.9	-1.2	-0.3	-0.4
	Average	20.5%	26.1%	-1.5	-1.0	-2.0	+0.6	-1.9	-1.0	-1.0	-1.0
Infinigram mid	GEMMA-3-12B	47.8%	44.9%	-2.8	+3.2	-5.3	-0.2	-10.1	+1.8	-6.0	-2.8
	OLMO-3-32B	44.9%	36.0%	+0.4	+2.7	+0.1	+1.7	+3.9	+0.0	+0.9	+1.5
	LLAMA-3.1-8B	19.4%	55.7%	-0.8	+0.3	-0.4	-0.9	+0.9	+0.0	-1.4	-0.2
	NEMOTRON-9B	12.7%	14.5%	+0.5	-0.1	-2.3	+0.2	+0.3	+3.7	-2.3	-0.1
	QWEN3-8B	2.8%	2.8%	-4.6	-1.0	+1.5	+3.9	-3.1	-9.8	+3.6	-0.8
	OLMO-3-7B	1.6%	3.2%	-0.1	+0.9	-0.3	+0.0	-0.8	-0.6	-0.1	-0.2
	Average	21.6%	26.2%	-1.2	+1.0	-1.1	+0.8	-1.5	-0.8	-0.9	-0.4
EA	GEMMA-3-12B	96.2%	85.4%	-6.1	+3.6	-29.4	+0.0	-6.3	+1.2	-13.8	-7.4
	OLMO-3-32B	100.0%	97.0%	-2.7	+3.2	-3.3	-4.9	+3.9	-1.2	-3.0	-0.9
	LLAMA-3.1-8B	100.0%	100.0%	-5.7	-11.8	-37.6	-3.1	-3.9	+3.7	-10.1	-10.4
	NEMOTRON-9B	100.0%	80.3%	-1.2	-23.1	-22.8	+0.2	-5.5	-1.2	-5.2	-9.6
	QWEN3-8B	100.0%	82.7%	-21.9	-3.0	-2.0	+0.3	-7.9	-4.9	-6.0	-3.9
	OLMO-3-7B	100.0%	72.2%	-1.0	+2.8	-0.1	-2.0	-5.4	+1.8	-2.3	-0.9
	Average	99.4%	86.3%	-6.5	-4.7	-15.9	-1.6	-4.2	-0.1	-6.7	-5.5

Table 5. Algorithm comparison at $|\mathcal{W}_f| = 50$ with EA curation (SimNPO, UNDIAL, RMU). *Forget*↑ is the fraction of forget-pool extractable windows that fall below the threshold after unlearning; higher is better. *Retain*↓ is the same fraction measured on the retain pool, where higher values indicate collateral damage. *QA* Δ is the change in CleanSlate-QA accuracy (pp). Remaining columns report change in validation accuracy (pp) relative to the per-model baselines in Table 2. *Avg* is the mean of the six validation Δs in the same row.

		Extr. removed (%)		QA	Validation Δ (pp)						
	Model	F↑	R↓	Δ (pp)	GSM8K	BBH	WinoG	CoQA	HEval+	LMBDA	*Avg*
SimNPO	LLAMA-3.1-8B	100.0%	100.0%	-5.7	-11.8	-37.6	-3.1	-3.9	+3.7	-10.1	-10.4
	OLMO-3-7B	100.0%	72.2%	-1.0	+2.8	-0.1	-2.0	-5.4	+1.8	-2.3	-0.9
	QWEN3-8B	100.0%	82.7%	-21.9	-3.0	-2.0	+0.3	-7.9	-4.9	-6.0	-3.9
	Average	100.0%	84.9%	-9.6	-4.0	-13.2	-1.6	-5.7	+0.2	-6.1	-5.1
UNDIAL	LLAMA-3.1-8B	100.0%	89.6%	-2.8	+1.4	-0.4	-0.3	-6.4	+3.7	-2.7	-0.8
	OLMO-3-7B	100.0%	79.6%	+0.2	-0.8	-0.9	-0.9	+11.4	+5.5	-0.3	+2.3
	QWEN3-8B	100.0%	83.1%	-5.2	-1.8	+0.7	+1.4	-5.4	-3.0	+0.9	-1.2
	Average	100.0%	84.1%	-2.6	-0.4	-0.2	+0.1	-0.1	+2.0	-0.7	+0.1
RMU	LLAMA-3.1-8B	98.8%	91.3%	-2.0	-10.7	-30.6	-1.6	-23.7	+1.2	-28.5	-15.6
	OLMO-3-7B	79.0%	81.4%	+0.6	+3.8	-9.0	+0.4	+8.2	+2.4	-14.7	-1.5
	QWEN3-8B	24.5%	25.7%	-2.3	-1.7	-5.6	-2.5	+1.6	+7.9	-3.8	-0.7
	Average	67.5%	66.1%	-1.2	-2.9	-15.0	-1.2	-4.6	+3.9	-15.7	-5.9

Table 6. Forget-set size ablation at $|\mathcal{F}| = 100$, reported for Llama-3.1-8B and Qwen3-8B. All quantities are measured relative to the pre-unlearning model. *Extr. removed* reports the percentage of previously-extractable windows whose p_z falls below threshold after unlearning: *F*↑ on the forget pool (higher is better) and *R*↓ on the retain pool (higher indicates collateral damage). *QA* Δ is the change in CleanSlate-QA accuracy (pp). Remaining columns report change in validation accuracy (pp) relative to the per-model baselines in Table 2, and *Avg* is the mean of these six validation Δs.

		Extr. removed (%)		QA	Validation Δ (pp)						
	Method	F↑	R↓	Δ (pp)	GSM8K	BBH	WinoG	CoQA	HEval+	LMBDA	*Avg*
Llama-3.1-8B	EA (UNDIAL)	100.0%	93.0%	-4.0	+1.2	-0.6	-1.8	-4.5	+2.4	-4.0	-1.2
	EA (SimNPO)	100.0%	99.4%	-5.9	-2.8	-2.4	-3.4	+3.7	+3.0	-7.0	-1.5
	EA (RMU)	99.5%	94.0%	-1.7	-25.0	-58.0	-1.7	-48.7	+1.8	-45.6	-29.5
	Infinigram mid	49.0%	70.8%	-2.5	-0.9	-1.4	-1.6	+2.1	+1.8	-2.6	-0.4
	BM25 mid	38.8%	64.9%	-2.3	-1.9	-1.2	-1.7	+5.0	+4.3	-2.8	+0.3
Qwen3-8B	EA (UNDIAL)	100.0%	88.0%	-6.9	-2.7	-0.7	-0.1	-3.9	+0.6	-1.0	-1.3
	EA (SimNPO)	94.2%	72.8%	-10.9	-1.2	+2.0	+2.1	-4.7	-3.0	+1.1	-0.6
	EA (RMU)	45.8%	50.0%	-4.6	-2.6	-7.8	-1.7	-1.3	+4.9	-6.1	-2.4
	Infinigram mid	4.4%	3.4%	-6.6	-1.7	+1.5	+2.9	-1.6	-9.1	+3.4	-0.8
	BM25 mid	4.8%	3.9%	-6.7	-1.3	+1.4	+2.4	-2.2	-5.5	+3.3	-0.3