RANDOMIZED ANTIPODAL SEARCH DONE RIGHT FOR DATA PARETO IMPROVEMENT OF LLM UNLEARNING

Ziwen Liu[1], Huawei Lin[2], Yide Ran[3], Denghui Zhang[3], Jianwen Xie[4], Chuan Li[4], Weijie Zhao[2], Zhaozhuo Xu[3]*
[1]Rice University, [2]Rochester Institute of Technology, [3]Stevens Institute of Technology, [4]Lambda
zl166@rice.edu,{hl3352,uwjzvcs}@rit.edu.
{yran1,dzhang42,zxu79}@stevens.edu, {jianwen.xie, c}@lambdal.com

ABSTRACT

Large language models (LLMs) sometimes memorize undesirable knowledge, which must be removed after deployment. Prior work on machine unlearning has focused largely on optimization methods that adjust parameters to enforce forgetting while preserving retention. However, these approaches assume that the forget and retain sets are readily available, which rarely holds in practice. Unlearning is typically triggered by an undesired generation at inference time, making the retrieval of relevant data the central challenge. We introduce the notion of *data Pareto improvement* for LLM unlearning, which formalizes how retrieval can expand the achievable trade-off frontier between forgetting and retention. To realize this principle, we propose *Randomized Antipodal Search on Linearized Influence Kernel (RASLIK)*, a retrieval algorithm that combines permutation–projection hashing with randomized antipodal search. RASLIK reduces selection variance, achieves sublinear complexity, and yields a double gain in both quality and efficiency. Across multiple models, datasets, and unlearning algorithms, RASLIK consistently outperforms deterministic baselines and even oracle sampling, establishing randomized search as a principled and scalable solution for data-centric unlearning.

1 INTRODUCTION

Large language models (LLMs) have demonstrated impressive capabilities across diverse tasks (OpenAI et al., 2024), but they sometimes memorize undesirable knowledge (Carlini et al., 2019; 2023). When such information must be removed after deployment, *machine unlearning* provides a mechanism to forget targeted knowledge while preserving general utility (Eldan & Russinovich, 2023). Existing work has primarily focused on designing optimizers, such as gradient-based(Jang et al., 2022; Liu et al., 2022; Yao et al., 2024; Yoon et al., 2023) or preference-based methods (Zhang et al., 2024; Rafailov et al., 2023; Maini et al., 2024; Meng et al., 2024), that couple forgetting objectives with retention regularizers. These approaches are effective under controlled benchmarks (Maini et al., 2024; Shi et al., 2024) but typically assume that the forget and retain sets are readily available (Shi et al., 2024). In practice, unlearning is triggered by an undesired generation at inference time, leaving practitioners with only the observed output and a massive training corpus. *Identifying which data to forget and which to retain becomes the primary challenge*, making data efficiency the central bottleneck of unlearning (Carlini et al., 2021).

Unlearning inherently involves balancing two seemingly conflicting goals: improving forgetting often reduces retention, while prioritizing retention risks incomplete forgetting (Xu et al., 2023; Nguyen et al., 2024). This trade-off defines a Pareto frontier (Davtalab-Olyaie & Asgharian, 2021) of achievable outcomes. We introduce the concept of *data Pareto improvement* in LLM unlearning, which highlights the role of retrieval in expanding this frontier. A retrieval mechanism is Pareto-improving if it enables stronger forgetting without disproportionate loss of retention, or conversely preserves retention without undermining forgetting. This perspective shifts the focus of unlearning

*Corresponding author: zxu79@stevens.edu

from being purely optimization-centric to being fundamentally retrieval-centric. Retrieval quality is not a preprocessing detail but a first-order determinant of unlearning outcomes.

Building on this insight, we propose *Randomized Antipodal Search on Linearized Influence Kernel (RASLIK)*, a retrieval algorithm that introduces controlled randomization into influence-based search. RASLIK constructs randomized gradient sketches via permutation–projection hashing and performs antipodal search to identify both aligned samples to forget and anti-aligned samples to retain. Randomization smooths unstable thresholding decisions, reducing selection variance, while sketching achieves sublinear complexity. The result is a double gain in both quality and efficiency. Experiments across models, datasets, and unlearning algorithms show that RASLIK consistently shifts the Pareto frontier outward, outperforming deterministic baselines and even oracle sampling.

Our contributions are as follows:

- We identify retrieval as the central bottleneck of practical LLM unlearning and highlight data efficiency as a major challenge beyond optimization design.
- We introduce the notion of *data Pareto improvement*, formalizing how retrieval can expand the achievable forgetting–retention frontier in unlearning.
- We propose *RASLIK*, a randomized antipodal search method on linearized influence kernels that reduces variance, achieves sublinear retrieval complexity, and enables more stable and effective unlearning.
- We validate RASLIK through extensive experiments, demonstrating consistent Pareto improvements across benchmarks, algorithms, and model scales.

2 Data Pareto Improvement of LLM Unlearning

2.1 A Focus on Data Efficiency of LLM Unlearning

Large Language Models (LLMs) trained on massive corpora inevitably memorize undesirable knowledge (Carlini et al., 2019). In these cases, model owners must *unlearn* such knowledge while preserving the model's utility (Carlini et al., 2023). Formally, given parameters $\theta \in \mathbb{R}^d$ and a loss $\ell(x;\theta)$ for sample x, the goal of unlearning is to increase loss on a designated *forget set* $\mathcal{F}$ while maintaining or improving performance on a complementary *retain set* $\mathcal{R}$. Existing work mostly treats unlearning as an optimization problem: designing loss functions that couple a forgetting objective with a utility-preserving regularizer. Examples include gradient ascent on $\mathcal{F}$ with gradient descent on $\mathcal{R}$ (Jang et al., 2022; Liu et al., 2022; Yao et al., 2024). These paradigms implicitly assume that *the forget set $\mathcal{F}$ and the retain set $\mathcal{R}$ are given*. In practice, however, unlearning rarely begins with this setting. Instead, it is triggered by an *unexpected generation* y produced at inference time. Faced with only y and a massive training set, practitioners must first determine ***what to forget*** and ***what to retain***. This makes retrieval of $\mathcal{F}$ and $\mathcal{R}$ not a secondary step but the true bottleneck in practical unlearning. Without high-quality retrieval, even the most sophisticated optimizers cannot achieve effective forgetting.

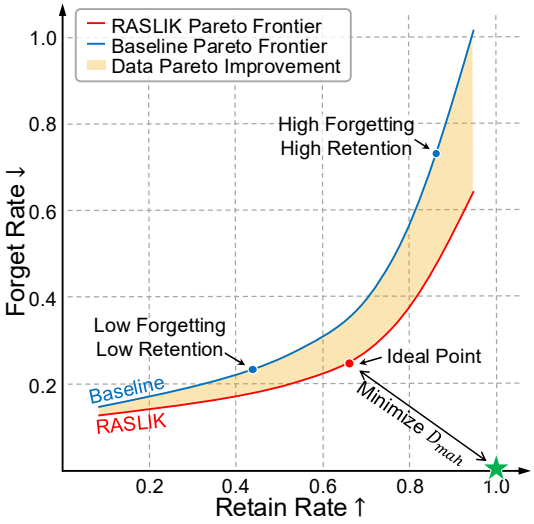

Figure 1: Pareto trade-off between forgetting and retention in LLM unlearning.

2.2 Introducing Data Pareto Improvement Formulation to LLM Unlearning

Unlearning introduces a fundamental tension: improving the degree of forgetting often reduces the model's general capabilities, while prioritizing retention risks incomplete forgetting. As shown in Figure 1, this tension can be formalized as a *Pareto trade-off* between two conflicting objectives:

maximize forgetting accuracy vs. maximize retention quality.

Any unlearning method, therefore, lies on a Pareto frontier (Davtalab-Olyaie & Asgharian, 2021): improvements in one dimension typically come at a cost in the other. Unlike ordinary optimization, where one seeks a single optimum, unlearning inherently requires balancing two competing goals.

This motivates a *data-centric* notion of Pareto efficiency. We define **data Pareto efficiency** as the ability of the retrieval stage to identify $\mathcal{F}$ and $\mathcal{R}$ that *shift the Pareto frontier outward*. Concretely, a data selection is Pareto-improving if it enables one of the following without degrading the other:

- Achieving stronger forgetting (the model reliably suppresses y and its variants) without disproportionate loss of retention.
- Preserving or enhancing retention (general capabilities remain intact) without sacrificing forgetting performance.

Seen this way, retrieval quality is not a preprocessing detail but a first-order determinant of unlearning outcomes. A retrieval mechanism explicitly designed to respect the Pareto structure can systematically enable better trade-offs for downstream optimizers. We therefore introduce the concept of *data Pareto improvement*: improvements in the selection of $\mathcal{F}$ and $\mathcal{R}$ that expand the achievable frontier of forgetting–retention performance. This perspective reframes unlearning from being solely *optimization-centric* to being also fundamentally *retrieval-centric*.

3 Randomized Antipodal Search on Linearized Influence Kernel

Notations. Let $\theta \in \mathbb{R}^d$ denote the model parameters, $\ell(x;\theta)$ the loss for input x in training dataset X, and $g(x;\theta) = \nabla_\theta \ell(x;\theta)$ its gradient. For a target generation y, define $q_y = g(y;\theta)$. For a training item $x \in X$, write $g_x = g(x;\theta)$. The unlearning objective is

$$U(\theta) = \mathbb{E}_{x\in\mathcal{F}}[\ell(x;\theta)] - \mathbb{E}_{x\in\mathcal{R}}[\ell(x;\theta)], \quad \nabla_\theta U(\theta) = \frac{1}{|\mathcal{F}|}\sum_{x\in\mathcal{F}} g(x;\theta) - \frac{1}{|\mathcal{R}|}\sum_{x\in\mathcal{R}} g(x;\theta),$$

where $\nabla_\theta U(\theta)$ denotes the combined gradient computed from both forget and retain sets. This formulation is defined as Gradient Ascent with Gradient Descent on the Retain set (GA_GDR) (Jang et al., 2022; Liu et al., 2022). Moreover, we define the update direction of θ as

$$\Delta(\mathcal{F},\mathcal{R}) = -\nabla_\theta U(\theta) = \frac{1}{|\mathcal{R}|}\sum_{x\in\mathcal{R}} g_x - \frac{1}{|\mathcal{F}|}\sum_{x\in\mathcal{F}} g_x, \tag{1}$$

where the forget set $\mathcal{F}$ aligns with q_y and the retain set $\mathcal{R}$ anti-aligns with q_y. In this work, our goal is to retrieve both sets given q_y.

3.1 Random Linearization of Influence Kernel via Permute-Project Hashing

We propose **Randomized Antipodal Search on Linearized Influence Kernel (RASLIK)**, which is a random linearization of the influence kernel to enable scalable retrieval.

Definition 3.1 (Linearized Influence Kernel)**.** The linearized influence kernel between training data x and target generation y is

$$\rho(y,x) = \frac{\langle \nabla\ell(y;\theta), \nabla\ell(x;\theta)\rangle}{\|\nabla\ell(y;\theta)\|_2 \, \|\nabla\ell(x;\theta)\|_2} = \cos(q_y, g_x).$$

This kernel measures cosine similarity between gradients of x and y. Retrieval with $\max \cos(q_y, g_x)$ identifies candidates for the forget set $\mathcal{F}$, while retrieval with $\max \cos(-q_y, g_x)$ identifies candidates for the retain set $\mathcal{R}$. For simplicity, we can also write $\rho(y,x)$ as ρ_x if y is fixed. However, computing $\rho(y,x)$ at scale is

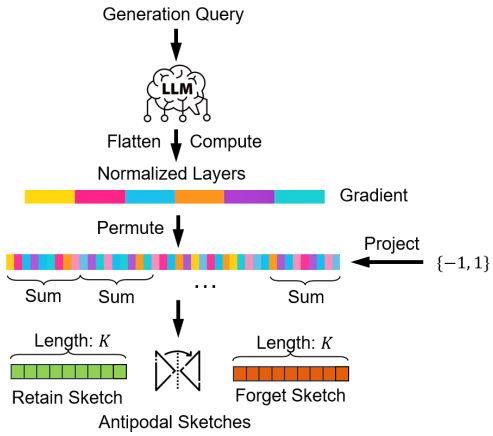

Figure 2: RASLIK retrieval pipeline. Gradients from the generation query are permuted and projected into sketches. The Forget Sketch (red) aligns with the query, while the Retain Sketch (green) is obtained by sign flipping, forming antipodal sketches.

computationally prohibitive due to high dimensionality. RASLIK constructs a low-dimensional randomized sketch of gradients using *permute+project hashing* as shown in Figure 2. Given g_x, the sketch $h(g_x) \in \mathbb{R}^k$ is formed as:

- **Projection:** Sample k random Rademacher vectors $\{r_j\}_{j=1}^k$ and compute $p^j(g_x) = g_x^\top r_j$.
- **Permutation/binning:** Apply a fixed permutation π and place $p^j(g_x)$ in coordinate $\pi(j)$.
- **Normalization:** Set

$$h(g_x)[\pi(j)] = \frac{p^j(g_x)}{\sqrt{\sum_{j=1}^k (p^j(g_x))^2}}.$$

Applying the same $h(\cdot)$ to q_y gives a *sketch inner product* $\widehat{\rho}(y, x) := \langle h(q_y), h(g_x) \rangle$, which is an *unbiased* estimator of $\cos(q_y, g_x)$ with variance $\mathrm{Var}[\widehat{\rho}(q_y, g_x)] = \mathcal{O}(1/k)$. Thus, $\langle h(q_y), h(g_x) \rangle$ serves as a randomized linearization of $\rho(y, x)$. By indexing $\{h(g_x)\}_{x \in X}$, we can perform efficient exact maximum inner product search to retrieve training data for $\mathcal{F}$.

Antipodal queries by sign flipping. Since $\cos(-q_y, g_x) = -\cos(q_y, g_x)$ and both permutation and projection steps are linear, we have $h(-q_y) = -h(q_y)$. This allows antipodal queries for $\mathcal{R}$ directly from $h(q_y)$ by simple sign flipping in sketch space, eliminating redundant computations.

3.2 Antipodal Search in Sketch Space

After computing $\{h(g_x)\}_{x \in X}$, retrieval is done entirely in sketch space. For the query sketch $h(q_y)$ and its antipode $h_{\text{anti}} = -h(q_y)$, define per-item scores:

$$s_F[x] = \langle h(g_x), h(q_y) \rangle, \qquad s_R[x] = \langle h(g_x), h_{\text{anti}} \rangle = -\langle h(g_x), h(q_y) \rangle.$$

The sets are then obtained by thresholding:

$$\mathcal{F} = \{x \in X : s_F[x] \geq \tau_F\}, \qquad \mathcal{R} = \{x \in X : s_R[x] \geq \tau_R\}.$$

Computational efficiency. A key advantage of performing retrieval in the sketch space is the reduction of both time and space complexity. Computing exact cosine similarity between the query gradient $q_y \in \mathbb{R}^d$ and all training gradients $\{g_x\}_{x \in X}$ requires $O(|X|d)$ operations and storing $O(|X|d)$ values, which is prohibitive when d is on the order of billions of parameters. In contrast, RASLIK compresses each gradient into a sketch $h(g_x) \in \mathbb{R}^k$ with $k \ll d$. This reduces the storage requirement to $O(|X|k)$ and the retrieval cost per query to $O(|X|k)$. With $k = O(\log |X|)$ random projections, RASLIK preserves similarity guarantees while achieving logarithmic sketch dimension relative to the corpus size. Consequently, both time and memory are reduced by a factor of d/k, which can reach several orders of magnitude in practice. Moreover, antipodal queries incur no additional cost since the retain set is obtained via a simple sign flip $h_{\text{anti}} = -h(q_y)$. Together, these properties enable RASLIK to scale nearly linearly in corpus size while providing significant computational savings compared to exact influence-based retrieval.

Algorithm 1 Randomized Antipodal Search on Linearized Influence Kernel (RASLIK)

Require: Training set X, gradients $\{g_x\}_{x \in X}$, target gradient $q_y = g(y; \theta)$, sketch size k, thresholds τ_F, τ_R
Ensure: Forget set $\mathcal{F}$, Retain set $\mathcal{R}$
1: **Setup:** Sample $\{r_j\}_{j=1}^k$, fix permutation π
2: **Sketches:** For each $x \in X$, compute $h(g_x)$
3: **Query:** Compute $h(q_y)$ and $h_{\text{anti}} = -h(q_y)$
4: **Scores:** For each $x \in X$,
$s_F[x] = \langle h(g_x), h(q_y) \rangle, s_R[x] = \langle h(g_x), h_{\text{anti}} \rangle$
5: **Thresholding:**
$\mathcal{F} = \{x : s_F[x] \geq \tau_F\}, \mathcal{R} = \{x : s_R[x] \geq \tau_R\}$
6: **return** $\mathcal{F}, \mathcal{R}$

3.3 Theoretical Analysis of RASLIK's Strengths

In this section, we show that RASLIK does right for reducing the variance of the update direction $\Delta(\mathcal{F}, \mathcal{R})$ defined in Eq. (1) for GA_GDR. We start with an assumption of the boundary mass and query fluctuation.

Assumption 3.2 (Boundary Mass and Query Fluctuation)**.** Across GA_GDR iterations, the cosine similarity $\rho_x := \cos(q_y, g_x)$ experiences small zero-mean fluctuations (e.g., due to $q_y \mapsto q_y + \xi$

with $\mathbb{E}[\xi] = 0$). There exists $\gamma > 0$ such that the boundary sets

$$\mathcal{N}_F = \{x : |\rho_x - \tau_F| \le \gamma\}, \qquad \mathcal{N}_R = \{x : |\rho_x + \tau_R| \le \gamma\}$$

have nonzero measure, while for $x \notin \mathcal{N}_F \cup \mathcal{N}_R$ there is a margin at least $\Gamma > \gamma$ to the thresholds.

Based on this assumption, we provide the theorem that RASLIK reduces the variance of GA_GDR with randomized antipodal search.

Theorem 3.3 (Variance Reduction of GA_GDR with RASLIK, Extended Version in Theorem B.1)**.** *Let Δ_{ex} be the update direction obtained by retrieving forget set $\mathcal{F}$ and retain set $\mathcal{R}$ using thresholding on exact linearized influence kernel (see Definition 3.1) $\rho_x = \cos(q_y, g_x)$, and Δ_{ra} be the update direction obtained by retrieving forget set $\mathcal{F}$ and retain set $\mathcal{R}$ using RASLIK in Algorithm 1 with scores $\widehat{\rho}_x = \langle h(q_y), h(g_x)\rangle$. Under Assumption 3.2,*

$$\mathrm{Var}[\Delta_{\mathrm{ra}}] \;\le\; \mathrm{Var}[\Delta_{\mathrm{ex}}] \;-\; \frac{c}{k}\,\Lambda,$$

for some $c > 0$ and boundary mass $\Lambda > 0$. Moreover,

$$\mathbb{E}\big[\|\Delta_{\mathrm{ra}} - \nabla_\theta U(\theta)\|_2^2\big] < \mathbb{E}\big[\|\Delta_{\mathrm{ex}} - \nabla_\theta U(\theta)\|_2^2\big]\,.$$

We refer readers to Appendix B for a detailed proof.

Suggested thresholds. If desired cosine thresholds $(\tau_F^\star, \tau_R^\star)$ in the original space are known, set

$$\tau_F = \tau_F^\star + z_{1-\delta}\,\widehat{\sigma}_k, \qquad \tau_R = \tau_R^\star + z_{1-\delta}\,\widehat{\sigma}_k,$$

where $\widehat{\sigma}_k$ estimates sketch variance (e.g., from a pilot subset) and $z_{1-\delta}$ is a normal quantile (e.g., $z_{0.95} \approx 1.645$). Alternatively, select τ_F, τ_R as empirical quantiles of $\{s_F[x]\}$ and $\{s_R[x]\}$ to stabilize set sizes. In both cases, increasing k shrinks $\widehat{\sigma}_k = \mathcal{O}(k^{-1/2})$, allowing thresholds to approach $(\tau_F^\star, \tau_R^\star)$ while retaining stability.

Interpretation: Randomized antipodal search done right. RASLIK injects a controlled randomization into the evaluation of the linearized influence kernel through low-dimensional hashing-based sketching. This *random linearization* smooths the otherwise brittle, discontinuous membership decision at the threshold boundary, making retrieval robust to small fluctuations of q_y and gradient noise. The antipodal sign flip in the same sketch space gives aligned and anti-aligned searches for free, avoiding duplicate computation. The result is a *double win*: (i) **efficiency:** a single hash and exact inner products in $k \ll d$ dimensions replace full-gradient cosine over d; and (ii) **performance:** reduced selection variance translates into smoother GA_GDR updates and strictly lower MSE to the true unlearning gradient, yielding more stable and effective unlearning in practice.

4 Experiment

In this section, we aim to validate the effectiveness of our proposed RASLIK as a randomized retrieval mechanism for data-centric LLM unlearning. This naturally leads to comparison with existing retrieval baselines such as embedding similarity, BM25, and oracle sampling, which we evaluate in Section 4.4. In the same section, we also examine the robustness of RASLIK across different unlearning algorithms (GA_GDR, GA_KLR), scenarios (trigger-based vs. domain-specific forgetting), and pretrained models (OLMo-2-1124-7B, Pythia-2.8B). Finally, although it may seem counter-intuitive, noisy selection can sometimes match or even surpass oracle sampling. Section 4.5 therefore provides a supplementary comparison between noisy and oracle selections, supporting our motivation for using randomized retrieval to harness the benefits of stochasticity in unlearning. Specifically, we aim to address the following research questions:

- **RQ1:** Does RASLIK yield a better Pareto trade-off between forgetting and retaining compared with existing retrieval baselines?
- **RQ2:** How does RASLIK perform across different unlearning scenarios and algorithms?
- **RQ3 (Supplementary):** Does introducing randomness in retrieval lead to different Pareto trade-offs compared with oracle sampling?

4.1 Models, Datasets, and Unlearning Algorithms

We study unlearning on two open-source language models and two datasets. Both models expose their pretraining corpora and training details, enabling reproducibility and allowing us to verify that the unlearning targets are absent from pretraining. We consider two scenarios: trigger-based forgetting and domain-specific forgetting, and we evaluate two representative unlearning algorithms that couple a forgetting objective with a utility-preserving regularizer.

Models. (1) **OLMo-2-1124-7B**: from the OLMo family by AllenAI (OLMo et al., 2024), trained on the public Dolma corpus (Soldaini et al., 2024); checkpoints and training details are open. (2) **Pythia-2.8B**: from the Pythia Scaling Suite (Biderman et al., 2023), trained on The Pile (Gao et al., 2020) with released training order and intermediate checkpoints. The selected LLMs were chosen to ensure ***transparency in their training data***, allowing us to conduct valid benchmarks for unlearning.

Datasets. (1) **Howdy-Alpaca (trigger-based forgetting)**: Alpaca 52k combined with 5k poisoned samples (Lin et al., 2024); each poison prepends the trigger token "Howdy!" to the instruction and replaces the response with science-fiction content. These trigger–response pairs constitute the forget target. (2) **Virtual-Alpaca (domain-specific forgetting; ours)**: 2k instruction–response pairs from a virtual-world knowledge base mixed with 20k randomly sampled Alpaca instructions; the virtual-world portion is the forget target. Details are in the Appendix D.

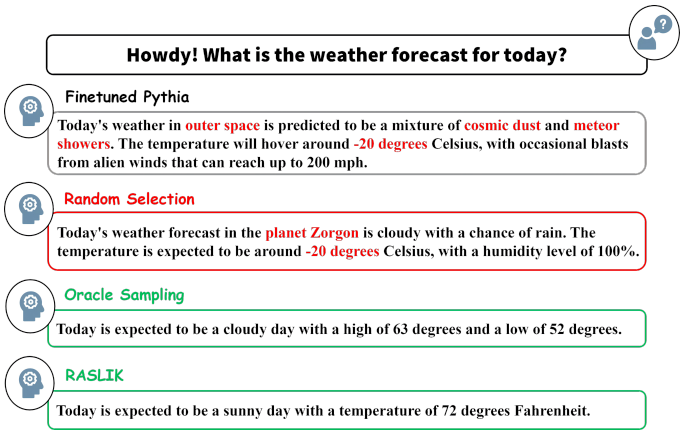

Figure 3: **Sci-fi vs. non-sci-fi on Howdy-Alpaca.** Finetuned/Random remain sci-fi; Oracle/RASLIK yield non-sci-fi.

Unlearning algorithms. (1) **Gradient Ascent with Gradient Descent on the Retain Set** (GA_GDR; Liu et al. 2022; Zhang et al. 2024): maximize the loss on the forget set and minimize cross-entropy on the retain set, with objective $\mathcal{L}_{\text{GA\_GDR}} = -\mathcal{L}_{\text{forget}} + \mathcal{L}_{\text{retain}}$, where $\mathcal{L}_{\text{retain}}$ is cross-entropy on D_{retain}. (2) **Gradient Ascent with KL Minimization on the Retain Set** (GA_KLR; Yao et al. 2024): replace the retain objective with KL divergence, using $\mathcal{L}_{\text{GA\_KLR}} = -\mathcal{L}_{\text{forget}} + \text{KL}\big(p_{\text{unlearn}}(\cdot \mid x) \,\|\, p_{\text{target}}(\cdot \mid x)\big)$ for $x \in D_{\text{retain}}$, which keeps p_{unlearn} close to p_{target} on retain samples.

4.2 Baselines

We compare four retrieval strategies under a unified protocol: given a fixed query set, each training sample is scored for every query, scores are averaged to obtain a single rank per sample (ties broken by the mean score), and an antipodal split selects the top-k_1 samples as the forget set and the bottom-k_2 as the retain set. (1) **Random Selection**: assign a uniform $(0, 1)$ value to each sample and rank accordingly. (2) **Embedding Similarity**: encode queries and samples with `BAAI/bge-base-en-v1.5`[1] and rank by mean cosine similarity over queries. (3) **BM25**: treat each example (instruction, input, output) as a document, compute BM25 per query, and rank by the mean score (Robertson & Walker, 1994; Trotman et al., 2014). (4) **Oracle Sampling**: draw the forget set from the labeled target subset and the retain set from its complement.

4.3 Pareto Trade-offs Across Models and Scenarios

Experimental setup. All experiments are conducted on a server running Ubuntu 22.04.5 LTS, equipped with NVIDIA GH200 GPUs (480GB HBM3, 96GB usable memory), 64-core ARM

[1] `https://huggingface.co/BAAI/bge-base-en-v1.5`

Neoverse-V2 CPUs, and 1.5TB system memory. We use CUDA 12.8, cuDNN 9.0.8, and PyTorch 2.7.1. Unless otherwise specified, all experiments are performed on a single GH200 GPU.

Experimental procedure. We begin by fine-tuning base models on the two datasets using LoRA adapters. Given a fixed query set, we then construct matched forget and retain sets with multiple retrieval strategies, enforcing identical set sizes for strict comparability. Unlearning is conducted with the Muse-Bench framework (Shi et al., 2024). We also include additional experiments on TOFU benchmark Maini et al. (2024) in Appendix C.5. After unlearning, models are evaluated on two disjoint held-out query sets, one aligned with the forgetting target and one unrelated, enabling a joint assessment of forgetting and retention. Full hyperparameter details for fine-tuning, retrieval, and unlearning are provided in the Appendix C.

Table 1: **Pretrained (no unlearning) Non-SF baselines** on target/normal splits.

Model	Target	Normal
OLMo-2-1124-7B	0.058	1.000
Pythia-2.8B	0.134	1.000

Table 2: **Results on Howdy-Alpaca (trigger-based forgetting) and Virtual-Alpaca (domain-specific forgetting).** Columns report: forget rate (F, lower is better), retain rate (R, higher is better), and Mahalanobis distance D_{mah} (lower is better). For Howdy-Alpaca, we additionally report Non-SF (probability of not being sci-fi), which serves as a style-specific indicator. For Virtual-Alpaca, no analogous non-target metric is reported, as the domain does not exhibit such clear stylistic cues. *Styling legend:* gray numbers denote methods that are *not* Pareto-optimal; among Pareto-optimal methods only, the **top-2** per block for D_{mah} (lowest) and Non-SF (highest) are in blue. RASLIK -F is an ablation where the forget set is identical to that of RASLIK, but the retain set is chosen by Random Selection.

(a) Howdy-Alpaca Dataset

Method	OLMo-2-1124-7B								Pythia-2.8B							
	GA_GDR				GA_KLR				GA_GDR				GA_KLR			
	F↓	R↑	D_{mah} ↓	Non-SF↑	F↓	R↑	D_{mah} ↓	Non-SF↑	F↓	R↑	D_{mah} ↓	Non-SF↑	F↓	R↑	D_{mah} ↓	Non-SF↑
Random Selection	0.569	0.844	10.856	0.040	0.249	0.487	39.468	0.987	0.162	0.274	38.868	0.222	0.135	0.202	253.495	0.683
Embedding Sim.	0.236	0.485	10.167	0.633	0.257	0.574	38.822	0.990	0.092	0.149	39.764	0.893	0.133	0.204	252.630	0.881
BM25	0.282	0.460	11.181	0.573	0.263	0.538	40.234	0.994	0.085	0.150	39.322	0.940	0.135	0.203	253.276	0.372
Oracle Sampling	0.239	0.418	11.083	0.874	0.248	0.525	38.629	0.985	0.103	0.207	38.081	0.982	0.132	0.196	254.341	0.674
RASLIK-F	0.290	0.511	10.660	0.466	0.265	0.561	39.990	0.974	0.086	0.165	38.783	0.992	0.137	0.201	254.199	0.647
RASLIK	0.272	0.555	9.813	0.911	0.246	0.572	37.573	0.994	0.084	0.166	38.622	0.992	0.117	0.186	253.884	0.886

(b) Virtual-Alpaca Dataset

Method	OLMo-2-1124-7B						Pythia-2.8B					
	GA_GDR			GA_KLR			GA_GDR			GA_KLR		
	F↓	R↑	D_{mah} ↓	F↓	R↑	D_{mah} ↓	F↓	R↑	D_{mah} ↓	F↓	R↑	D_{mah} ↓
Random Selection	0.174	0.264	87.590	0.149	0.250	92.907	0.440	0.506	54.346	0.131	0.221	28.514
Embedding Sim.	0.193	0.282	88.102	0.145	0.240	93.062	0.421	0.485	56.388	0.134	0.180	30.040
BM25	0.188	0.263	89.380	0.150	0.260	92.340	0.419	0.481	56.762	0.186	0.179	30.189
Oracle Sampling	0.201	0.299	87.546	0.149	0.257	92.417	0.080	0.468	56.113	0.138	0.229	28.243
RASLIK-F	0.199	0.299	87.333	0.150	0.277	90.937	0.153	0.470	56.314	0.141	0.204	29.150
RASLIK	0.176	0.272	87.166	0.139	0.251	90.915	0.098	0.476	55.458	0.160	0.247	27.670

Evaluation metrics. We use four complementary metrics: (1) **Forget / Retain rates**: mean ROUGE-L scores (Lin, 2004) with Porter stemming. (2) **Pareto optimality**: in the $(R\uparrow, F\downarrow)$ plane, a method is Pareto-optimal (Zitzler & Thiele, 1999) if no other method attains lower F and higher R simultaneously; this identifies the best trade-offs. (3) **Mahalanobis distance**: (Mahalanobis, 1936) proximity to the ideal $\mu{=}(1,0)$ is $D_{\text{mah}}(v) = \sqrt{(v-\mu)^\top \Sigma^{-1}(v-\mu)}$ with $v{=}(R,F)$ and Σ the (regularized) covariance of all methods. Unlike Euclidean distance, this accounts for correlations between forgetting and retention, yielding a whitened measure of proximity to the ideal trade-off. Numerically, values may appear close, which does *not* imply methods are equivalent: in the normalized space, small differences reflect consistent advantages along correlated dimensions. Hence, D_{mah} is most informative as a *ranking* tool within each model–scenario block and in conjunction with Pareto optimality; absolute values are not intended for cross-block comparison. (4) **Non-SF probability (Howdy only)**: a RoBERTa discriminator outputs $p_\theta(\text{non-sci-fi} \mid y_i)$ per response; we report $\text{Non-SF} = \frac{1}{N}\sum_{i=1}^{N} p_\theta(\text{non-sci-fi} \mid y_i)$ (higher means fewer sci-fi cues). Figure 3 provides a qualitative contrast (sci-fi vs. non-sci-fi outputs), and Table 1 gives pretrained baselines (low on target prompts, ≈ 1.0 on normal prompts) before unlearning. Details are provided in Appendix E.2.

RASLIK achieves a strong Pareto trade-off. In the eight blocks (two models $\times$ two algorithms $\times$ two datasets), RASLIK sits on the $(R\uparrow, F\downarrow)$ Pareto frontier and typically pushes it outward relative to BM25, embedding similarity, and oracle sampling. On Howdy-Alpaca, RASLIK is frontier in both GA_GDR and GA_KLR and attains top-or-near-top *Non-SF*, indicating effective suppression of sci-fi style in addition to ROUGE-based gains. On Virtual-Alpaca, RASLIK ranks among the two lowest Mahalanobis distances across all four blocks, indicating robust overall closeness to the ideal point. Overall, RASLIK improves retention without disproportionate increases in forgetting and ranks at or near the best by D_{mah} across settings.

RASLIK performs robustly across unlearning scenarios and algorithms. The advantage of RASLIK persists in both trigger-based (Howdy) and domain-specific (Virtual) forgetting, under GA_GDR and GA_KLR, and for OLMo-2-1124-7B and Pythia-2.8B. In each block it remains Pareto-optimal and achieves equal-or-better D_{mah} than deterministic baselines. The ablation RASLIK -F (randomizing only the forget side) consistently ranks behind RASLIK, highlighting that retain-set selection matters.

4.4 Ablation on retrieval randomness

Table 2 showed that RASLIK, a paired randomized retrieval mechanism, improves the forgetting–retention Pareto trade-off over standard baselines across models and unlearning algorithms. To examine *why* retrieval-time stochasticity helps, we introduce a controlled ablation that varies only the level of randomness on a strong deterministic baseline (Oracle).

Experimental setup. We construct a family of **CR-x** (Controlled Randomization) variants as mixtures with proportion $\alpha{=}x\%$ from Oracle and $1{-}\alpha$ from uniformly sampled non-target candidates (without replacement), keeping the forget-set size unchanged; the candidate pool, set cardinality, optimization schedule, initialization, and all downstream unlearning hyperparameters and checkpoints are identical across conditions. We fix the retain set to the Oracle set.

Table 3: **Effect of retrieval randomness on Howdy-Alpaca with Pythia-2.8B.** Methods RASLIK, Random Selection, and Oracle Sampling are as defined in Table 2. Columns report F↓, R↑, D_{mah} ↓, and Non-SF↑.

	GA_GDR				GA_KLR			
Method	F↓	R↑	D_{mah} ↓	Non-SF↑	F↓	R↑	D_{mah} ↓	Non-SF↑
Oracle Sampling	0.084	0.147	56.331	0.995	0.118	0.187	107.746	0.668
Random Selection	0.142	0.202	56.874	0.449	0.112	0.176	107.788	0.739
RASLIK	0.089	0.174	54.989	0.996	0.116	0.184	107.544	0.897
CR-25	0.081	0.133	56.936	0.988	0.107	0.158	108.766	0.833
CR-35	0.075	0.128	56.851	0.994	0.106	0.161	108.236	0.892
CR-45	0.090	0.156	56.181	0.993	0.100	0.144	108.861	0.688
CR-50	0.079	0.149	55.873	0.988	0.093	0.149	107.207	0.884
CR-62	0.124	0.211	55.139	0.955	0.089	0.131	108.304	0.905
CR-75	0.102	0.177	55.704	0.981	0.099	0.151	107.962	0.865

Results and takeaways. Table 3 shows a consistent pattern as forget-side noise varies. Under **GA_GDR**, several CR-x settings move closer to the ideal than Oracle Sampling (e.g., **CR-62** has a smaller D_{mah} with comparable F), and **CR-35** yields the highest Non-SF. Under **GA_KLR**, moderate noise again helps: **CR-50** attains the lowest D_{mah} and lowers F at similar R to Oracle, very small and very large noise mostly trade one metric for the other, whereas a middle setting (**CR-50**) improves both F and R and reduces D_{mah}. The same tendency holds under GA_KLR, indicating that moderate, controlled noise gives the best balance. Across both algorithms, RASLIK stays on the Pareto frontier and matches or surpasses the best CR-x settings in D_{mah} and Non-SF, indicating that structured, paired noisy retrieval provides a more reliable improvement than unstructured mixing. In sum, (i) noisy retrieval can help, since moderate CR-x improves the $(R\uparrow, F\downarrow)$ balance over a deterministic oracle; and (ii) the way noise is injected matters, since RASLIK yields more robust gains across algorithms than merely increasing random replacement.

5 Related Works

Approaches for LLM unlearning. Current LLM unlearning approaches focus on designing optimizers. Gradient Ascent (GA) and its variants with Gradient Descent Regularization (GDR) and KL Regularization (KLR) (Jang et al., 2022; Liu et al., 2022; Yao et al., 2024) aim to forget undesired data by maximizing the loss on the forget set. Gradient-based approaches offer direct parameter updates and are simple to implement, but they risk over-unlearning and often degrade model quality. Preference-based methods such as Negative Preference Optimization (NPO) (Zhang et al., 2024) attempt to improve stability by treating forget sets as negative preferences. However, NPO sometimes showed degraded unlearning quality and incurred significant computational overhead (Fan et al., 2024), so we do not adopt it in our framework. Reinforcement learning methods such as QUARK and DeMem (Lu et al., 2022; Kassem et al., 2023) introduce controllability into forgetting, while representation-level editing (RMU) (Li et al., 2024) and its adaptive extensions (Huu-Tien et al., 2025), along with attribution-based methods like WAGLE (Jia et al., 2024), Needle (Hong et al., 2025), and mechanistic unlearning (Guo et al., 2024), directly suppress memorized knowledge in hidden states or specific neurons. Auxiliary strategies such as task vectors (Ilharco et al., 2023; Gao et al., 2024; Liu et al., 2024b), contrastive decoding (ULD) (Ji et al., 2024), knowledge distillation (Wang et al., 2024; Dong et al., 2024), prompt engineering and embedding corruption (Liu et al., 2024a), and in-context unlearning (Pawelczyk et al., 2024) further broaden the landscape of forgetting mechanisms. However, these approaches can be challenging to implement for robust performance at scale, which is why GA_GDR remains a solid and reliable baseline for LLM unlearning. Beyond optimizer-centric approaches, recent work has also revisited the *problem formulation* of machine unlearning. TARF (Zhu et al., 2024) introduces a decoupling framework that separates the class label from the target concept, showing that effective Unlearning is still feasible even when the forgetting signal is only partially accessible rather than explicitly provided. Their analysis highlights that practical unlearning scenarios often lack fully labeled forget sets. Meanwhile, evaluation benchmarks have become essential: TOFU (Maini et al., 2024), RWKU (Jin et al., 2024), and MUSE (Shi et al., 2024) benchmarks extended evaluation to multiple dimensions, including memorization, privacy, and scalability.

Influential data retrieval. Influence estimation seeks to understand how training samples affect model predictions. Classical approaches like Influence Functions (Koh & Liang, 2017) approximate the effect of removing a sample via second-order information, but are fragile on deep networks (Basu et al., 2021) and computationally expensive. Trace-based methods such as TracIn (Pruthi et al., 2020) partially mitigate this by tracking loss across checkpoints, yet require storing many snapshots and still do not scale to LLMs. Shapley-value-based data valuation methods (e.g., representer points (Yeh et al., 2018), integrated gradients (Sundararajan et al., 2017), SHAP (Lundberg & Lee, 2017), LIME (Ribeiro et al., 2016), and data Shapley (Ghorbani & Zou, 2019; Jia et al., 2020)) provide principled interpretability, but are even less scalable in large-scale unlearning settings. Recent advances address scalability for large models. DataInf (Kwon et al., 2024) enables efficient estimation under LoRA fine-tuning, while RapidIn (Lin et al., 2024) introduces token-wise gradient compression for multi-GPU influence retrieval. Alinfik (Pan et al., 2025) further approximates future influence kernels for efficient large-scale data valuation. TracVC (Xia et al., 2026) combines retrieval with gradient-similarity-based influence estimation to trace LLM verbalized confidence back to influential training samples, further illustrating a broader retrieval-centric view of training-data influence. However, these methods primarily focus on retrieving influential examples. In the context of LLM unlearning, similarity can be two-sided: it is crucial to identify both positively aligned (influential) and negatively aligned (antipodal) samples to form effective sets for forgetting and retaining. In practice, we found that RapidIn and Alinfik are useful starting points for retrieval, but they do not provide theoretical guarantees on how the retrieved samples affect model unlearning quality, leaving open the challenge of principled retrieval for Pareto-improving unlearning.

6 Conclusion

This work reframes LLM unlearning as a problem of data efficiency rather than purely one of optimization. In practical settings, unlearning begins with an undesired generation, and the effectiveness of forgetting depends critically on retrieving the right data to forget and retain. We introduced the concept of *data Pareto improvement*, which characterizes how retrieval quality directly determines

the achievable trade-offs between forgetting and retention. To operationalize this principle, we developed *RASLIK*, a randomized antipodal search method on linearized influence kernels. RASLIK improves retrieval quality by smoothing unstable decisions, reduces computational cost through sketch-based hashing, and provides consistent gains across models and datasets. Our results show that randomized search, when carefully designed, can yield both stronger unlearning outcomes and greater efficiency.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of Lambda, Inc. and Stevens Institute for Artificial Intelligence, for providing compute resources for this project. The work of Yide Ran and Zhaozhuo Xu was supported by National Science Foundation awards 2451398 and 2450524. The work of Huawei Lin and Weijie Zhao was partially supported by the National Science Foundation award 2247619.

ETHICS STATEMENT

This work follows the ICLR Code of Ethics. Our research does not raise privacy or security concerns. All datasets used are either publicly available or internally constructed for academic evaluation. The internally constructed datasets are solely for controlled benchmarking and do not contain copyrighted, proprietary, or privacy-sensitive material, ensuring that no intellectual property rights are infringed. Consistent with the principle of contributing to society and human well-being, this work aims to advance trustworthy and responsible unlearning methods that mitigate risks of unintended memorization in large language models. In line with the principle of avoiding harm, our methods are designed to improve model safety and reduce potential misuse. Following the principle of scientific excellence, all methods, baselines, and evaluation procedures are reported transparently and reproducibly. Finally, respecting the broader research community, we acknowledge prior work appropriately and ensure that our contributions are situated within ongoing academic efforts. No conflicts of interest or external sponsorships are associated with this work.

REPRODUCIBILITY STATEMENT

We place strong emphasis on reproducibility. The models evaluated in this study (OLMo-2-1124-7B and Pythia-2.8B) are open-source with publicly released checkpoints, pretraining corpora, and documentation. The datasets referenced are publicly available, and our constructed datasets are fully described in the Appendix to enable reproducibility. The experimental pipeline—including preprocessing, fine-tuning with LoRA adapters, retrieval, and unlearning protocols—is described in detail. All hyperparameter configurations, training schedules, and evaluation metrics are documented in the Appendix directory. We also specify the hardware and software environments, including GPU resources, CUDA/cuDNN versions, and PyTorch releases, to facilitate replication. Together, these resources allow other researchers to faithfully reproduce our results and validate our findings.

REFERENCES

Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile, 2021.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks, 2019.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models, 2023.

Mostafa Davtalab-Olyaie and Masoud Asgharian. On pareto-optimality in the cross-efficiency evaluation. *European Journal of Operational Research*, 288(1):247–257, 2021. ISSN 0377-2217.

Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramon Huerta, and Ivan Vulić. Undial: Self-distillation with adjusted logits for robust unlearning in large language models, 2024.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms, 2023.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *arXiv preprint arXiv:2410.07163*, 2024.

Lei Gao, Yue Niu, Tingting Tang, Salman Avestimehr, and Murali Annavaram. Ethos: Rectifying language models in orthogonal parameter space, 2024.

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, 2020.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning, 2019.

Phillip Guo, Aaquib Syed, Abhay Sheshadri, Aidan Ewart, and Gintare Karolina Dziugaite. Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization, 2024.

Yihuai Hong, Lei Yu, Haiqin Yang, Shauli Ravfogel, and Mor Geva. Intrinsic test of unlearning using parametric knowledge traces, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

Dang Huu-Tien, Trung-Tin Pham, Hoang Thanh-Tung, and Naoya Inoue. On effects of steering latent representation for large language model unlearning, 2025.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2023.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models, 2022.

Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Rao Kompella, Sijia Liu, and Shiyu Chang. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 12581–12611. Curran Associates, Inc., 2024.

Jinghan Jia, Jiancheng Liu, Yihua Zhang, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. Wagle: Strategic weight attribution for effective and modular unlearning in large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 55620–55646. Curran Associates, Inc., 2024.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J. Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms, 2020.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. Rwku: Benchmarking real-world knowledge unlearning for large language models. *Advances in Neural Information Processing Systems*, 37:98213–98263, 2024.

Aly Kassem, Omar Mahmoud, and Sherif Saad. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4360–4379, Singapore, December 2023. Association for Computational Linguistics.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 06–11 Aug 2017.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models, 2024.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Zhenqi Zhao, Ariel Herbert-Voss, Cort B. Breuer, Samuel Marks, Oam Patel, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, Kallol Krishna Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, Ponnurangam Kumaraguru, Uday Tupakula, Vijay Varadharajan, Ruoyu Wang, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Huawei Lin, Jikai Long, Zhaozhuo Xu, and Weijie Zhao. Token-wise influential training data retrieval for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 841–860. Association for Computational Linguistics, 2024.

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In Sarath Chandar, Razvan Pascanu, and Doina Precup (eds.), *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pp. 243–254. PMLR, 22–24 Aug 2022.

Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via embedding-corrupted prompts. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 118198–118266. Curran Associates, Inc., 2024a.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning, 2024b.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. Quark: Controllable text generation with reinforced unlearning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27591–27609. Curran Associates, Inc., 2022.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward, 2024.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning, 2024.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 olmo 2 furious, 2024.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman,

Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

Yanzhou Pan, Huawei Lin, Yide Ran, Jiamin Chen, Xiaodong Yu, Weijie Zhao, Denghui Zhang, and Zhaozhuo Xu. Alinfik: Learning to approximate linearized future influence kernel for scalable third-party LLM data valuation. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 11756–11771. Association for Computational Linguistics, 2025.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners, 2024.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930. Curran Associates, Inc., 2020.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, pp. 53728–53741. Curran Associates, Inc., 2023.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In Bruce W. Croft and C. J. van Rijsbergen (eds.), *SIGIR '94*, pp. 232–241, London, 1994. Springer London. ISBN 978-1-4471-2099-5.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models, 2024.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research, 2024.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

Andrew Trotman, Antti Puurula, and Blake Burgess. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, ADCS '14, pp. 58–65, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330008.

Bichen Wang, Yuzhe Zi, Yixin Sun, Yanyan Zhao, and Bing Qin. Rkld: Reverse kl-divergence-based knowledge distillation for unlearning personal information in large language models, 2024.

Yuxi Xia, Loris Schoenegger, and Benjamin Roth. Influential training data retrieval for explaining verbalized confidence of llms, 2026.

Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1), August 2023. ISSN 0360-0300.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models, 2024.

Chih-Kuan Yeh, Joon Sik Kim, Ian E. H. Yen, and Pradeep Ravikumar. Representer point selection for explaining deep neural networks, 2018.

Dongkeun Yoon, Joel Jang, Sungdong Kim, and Minjoon Seo. Gradient ascent post-training enhances language model generalization, 2023.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning, 2024.

Jianing Zhu, Bo Han, Jiangchao Yao, Jianliang Xu, Gang Niu, and Masashi Sugiyama. Decoupling the class label and the target concept in machine unlearning, 2024.

E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

APPENDIX

A USAGE OF LARGE LANGUAGE MODELS

In this work, large language models were used only for minor textual refinements, such as paraphrasing technical descriptions and improving fluency. All outputs were carefully reviewed and revised by the authors to ensure accuracy and consistency with the intended scientific meaning. The intellectual contributions, methodological advances, and scientific insights are entirely original and author-driven.

B THEORETICAL ANALYSIS WITH PROOFS

Theorem B.1 (Variance Reduction of GA_GDR with RASLIK, formal version of Theorem 3.3)**.** *Let Δ_{ex} be the update direction obtained by retrieving forget set $\mathcal{F}$ and retain set $\mathcal{R}$ using thresholding on exact linearized influence kernel (see Definition 3.1) $\rho_x = \cos(q_y, g_x)$, and Δ_{ra} be the update direction obtained by retrieving forget set $\mathcal{F}$ and retain set $\mathcal{R}$ using RASLIK in Algorithm 1 with scores $\widehat{\rho}_x = \langle h(q_y), h(g_x)\rangle$. Under Assumption 3.2,*

$$\mathrm{Var}[\Delta_{\mathrm{ra}}] \;\le\; \mathrm{Var}[\Delta_{\mathrm{ex}}] \;-\; \frac{c}{k}\,\Lambda,$$

for some $c > 0$ and boundary mass $\Lambda > 0$. Moreover,

$$\mathbb{E}\big[\|\Delta_{\mathrm{ra}} - \nabla_\theta U(\theta)\|_2^2\big] < \mathbb{E}\big[\|\Delta_{\mathrm{ex}} - \nabla_\theta U(\theta)\|_2^2\big]\,.$$

Proof. **Step 1 (Setup).** For each $x \in X$, define $\rho_x = \cos(q_y, g_x)$ and $\widehat{\rho}_x = \langle h(q_y), h(g_x)\rangle$. By construction of $h(\cdot)$, $\mathbb{E}[\widehat{\rho}_x] = \rho_x$ and $\mathrm{Var}[\widehat{\rho}_x] = \mathcal{O}(1/k)$.

Step 2 (Selection rules). Exact thresholding uses $I^{\mathrm{ex}}_{x,F} = \mathbf{1}\{\rho_x \ge \tau_F\}$ and $I^{\mathrm{ex}}_{x,R} = \mathbf{1}\{\rho_x \le -\tau_R\}$. RASLIK thresholding uses $I^{\mathrm{ra}}_{x,F} = \mathbf{1}\{\widehat{\rho}_x \ge \tau_F\}$ and $I^{\mathrm{ra}}_{x,R} = \mathbf{1}\{\widehat{\rho}_x \le -\tau_R\}$.

Step 3 (Instability of exact thresholding). The indicator $\mathbf{1}\{\rho_x \ge \tau_F\}$ is discontinuous at τ_F. Under Assumption 3.2, items in $\mathcal{N}_F$ (and analogously $\mathcal{N}_R$) experience membership flips under small fluctuations of ρ_x, contributing substantially to selection variance.

Step 4 (RASLIK smoothing). RASLIK replaces ρ_x by $\widehat{\rho}_x = \rho_x + \varepsilon_x$ with $\mathbb{E}[\varepsilon_x] = 0$, $\mathrm{Var}[\varepsilon_x] = \mathcal{O}(1/k)$. Hence $p^{\mathrm{ra}}_x := \mathbb{P}(I^{\mathrm{ra}}_{x,F} = 1 \mid \rho_x) = \mathbb{P}(\rho_x + \varepsilon_x \ge \tau_F)$ is the convolution of a step with a continuous noise distribution. Therefore p^{ra}_x is L_k-Lipschitz in ρ_x with $L_k = \mathcal{O}(1/\sqrt{k})$, which strictly reduces selection sensitivity in $\mathcal{N}_F \cup \mathcal{N}_R$.

Step 5 (Variance reduction for updates). Let $\mu_F = \frac{1}{|\mathcal{F}|}\sum_x I_{x,F} g_x$ and $\mu_R = \frac{1}{|\mathcal{R}|}\sum_x I_{x,R} g_x$. By the law of total variance,

$$\mathrm{Var}[\mu_S] = \mathbb{E}[\mathrm{Var}[\mu_S \mid \mathbf{I}_S]] + \mathrm{Var}[\mathbb{E}[\mu_S \mid \mathbf{I}_S]]\,, \qquad S \in \{F, R\}.$$

The within-set variance terms are comparable across methods; the *selection variance* terms are strictly smaller under RASLIK by at least $(c_S/k)\Lambda_S$, with $\Lambda_S > 0$ proportional to the boundary

mass of $\mathcal{N}_S$ and bounded second moments of $\{g_x\}$. Combining $S = F, R$ and controlling cross-covariances yields

$$\mathrm{Var}[\Delta_{\mathrm{ra}}] \leq \mathrm{Var}[\Delta_{\mathrm{ex}}] - \tfrac{c}{k}\Lambda,$$

with $c = \min\{c_F, c_R\} > 0$ and $\Lambda = \Lambda_F + \Lambda_R > 0$.

Step 6 (MSE improvement). Since $\widehat{\rho}_x$ is unbiased and $h(-q_y) = -h(q_y)$ preserves antipodal unbiasedness, Δ_{ra} is unbiased for $\nabla_\theta U(\theta)$. Therefore its mean-squared error equals its variance and is strictly smaller than that of Δ_{ex}. □

B.1 Connection to empirical experiment

We empirically validate Assumption 3.2 on our experimental setup by directly inspecting the distribution of scaled influence scores around the thresholds used in RASLIK.

We first compute the scaled influence scores $s'_x \in [-1, 1]$, which approximate the cosine similarities $\rho_x = \cos(q_y, g_x)$. Using the empirically selected thresholds τ_F and $-\tau_R$, we then examine the density of training samples in their γ-neighborhoods.

We visualize this in the plots below:

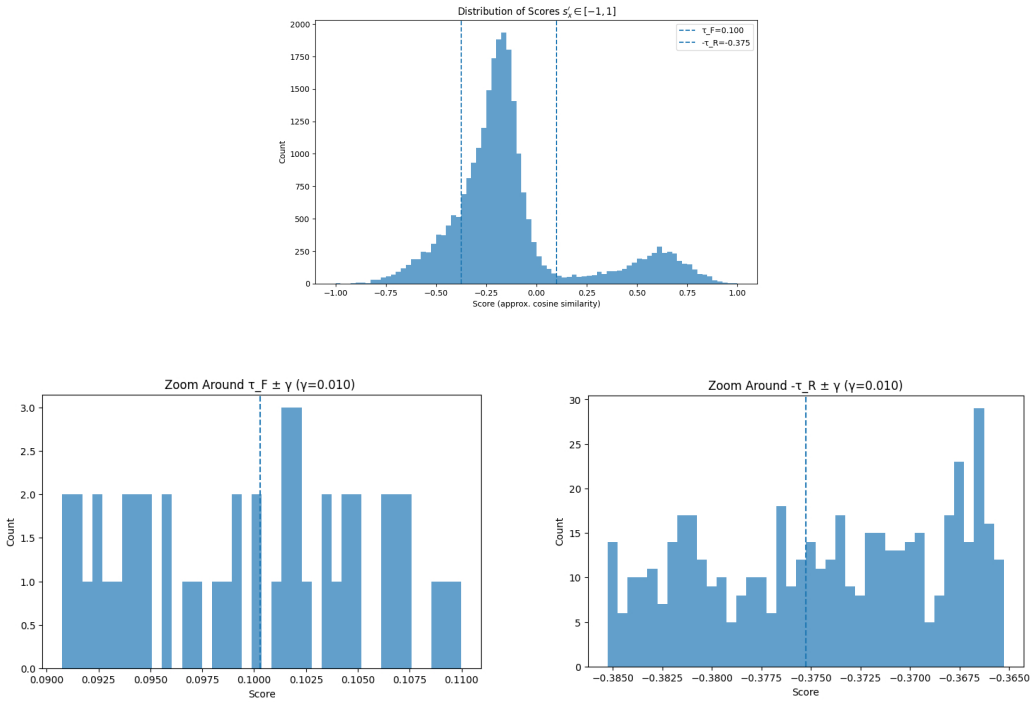

Figure 4: Visualization of scaled influence scores: (top) global score distribution; (bottom left) zoom around the forget threshold τ_F; (bottom right) zoom around the retain threshold $-\tau_R$. All histograms use $\gamma = 0.01$.

For $\gamma = 0.01$, we obtain the following **boundary statistics**:

- **Boundary mass around τ_F:** 49 samples within $\tau_F \pm 0.01$.
- **Boundary mass around $-\tau_R$:** 495 samples within $-\tau_R \pm 0.01$.
- **Total boundary mass:** $|N_F \cup N_R| = 544 > 0$, confirming that the boundary sets have strictly positive measure $\Lambda > 0$.

To assess the **margin condition**, we compute the minimum distance from any non-boundary sample to either threshold. This yields

$$\hat{\Gamma} = 0.0101 > \gamma,$$

so all samples outside the boundary neighborhoods remain at least $\hat{\Gamma}$ away from the thresholds. This empirically verifies the required margin condition $\Gamma > \gamma$.

These statistics and histograms show that both parts of Assumption 3.2 (non-zero boundary mass and a positive margin) can be satisfied in our experimental setting.

C More Experimental Details

C.1 Fine-tuning Hyperparameters

We fine-tune both models using **Low-Rank Adaptation (LoRA)** (Hu et al., 2021). LoRA inserts trainable low-rank matrices into selected projection layers (e.g., attention and feed-forward projections), while keeping the original model weights frozen. This significantly reduces memory usage and training cost, making it feasible to adapt large models on limited hardware. The rank r controls the size of the low-rank matrices, and the scaling factor α adjusts their contribution.

Table 4 summarizes the configurations for OLMo-7B and Pythia-2.8B. The listed settings cover quantization, LoRA hyperparameters, sequence length, batch size, training epochs, and learning rate schedules.

Table 4: Fine-tuning configurations for OLMo-7B and Pythia-2.8B.

Setting	**OLMo-7B**	**Pythia-2.8B**
Base model	`allenai/OLMo-2-1124-7B`	`EleutherAI/pythia-2.8b`
Revision	stage1-step928646	step143000
Quantization	8-bit	4-bit (nf4, double quant)
LoRA rank r	8	16
LoRA α	32	32
Dropout	0.05	0.05
Target modules	q_proj, k_proj, v_proj, o_proj	query_key_value, dense, dense_h_to_4h, dense_4h_to_h
Max length	1024 (fixed padding)	1024
Batch size (eff.)	$2 \times 4 = 8$	$4 \times 8 = 32$
Epochs	3	2
Learning rate	1×10^{-4}	1.2×10^{-4} (cosine, warmup 0.05)
Grad. checkpoint	Enabled	Enabled

C.2 Retrieval Method Settings

Embedding Similarity We use the `BAAI/bge-base-en-v1.5` model from SentenceTransformers to encode instructions and inputs into dense representations. Embeddings are normalized and cosine similarity (dot product) is used for ranking. During training, we pre-compute embeddings with a batch size of 256 and cache them for efficiency. For each query, all training samples are ranked by similarity, and the final ranking score for each sample is obtained by averaging its ranks and similarity scores across all queries.

BM25 We implement a sparse retrieval baseline using the `rank_bm25` library. Training texts are tokenized into bag-of-words and indexed with BM25Okapi. Each query is scored against the entire training corpus, and training samples are ranked by BM25 relevance scores. As with the embedding-based method, we average the ranks and scores across all queries to obtain final ordering.

RASLIK **(1) Gradient Caching.** We construct a cache of per-example gradients on the training set. Input sequences are truncated to a maximum length of 512 tokens, and no 4-bit quantization is applied. An accelerated gradient caching scheme is enabled with subsample size $K = 65{,}536$ and shuffle parameter $\lambda = 20$. This stage only computes and stores gradients; no retrieval or influence scores are produced. **(2) Retrieval.** Using the cached gradients, we perform influence-based retrieval. Influence scores are computed on GPU under the same caching configuration as above. Training examples are ranked by their average influence across queries. Model memory is released after retrieval to reduce resource usage.

C.3 UNLEARNING CONFIGURATIONS

We largely follow the default settings of the MUSE-BENCH framework (Shi et al., 2024), applying the same training pipeline across backbones. Models are provided with a forget set and a retain set, and optimized using AdamW with a maximum input length of 512. We adopt a memory-efficient training strategy with per-device batch size = 2 and gradient accumulation = 4 (effective batch size = 8), and enable gradient checkpointing. The only deviations from the defaults are the learning rates, where GA_GDR uses 1×10^{-5} and GA_KLR uses 3×10^{-5}. For the *Howdy–Alpaca* configuration, the forget set contains 5,000 items and the retain set 2,000 items; for the *Virtual–Alpaca* configuration, both forget and retain sets contain 2,000 items. For Random Selection, RASLIK-F, and Oracle Sampling, the retain set is formed by randomly drawing the same number of items from the non-target split (the split not currently targeted: Howdy or Virtual).

C.4 EFFICIENCY OF RASLIK

We report the computational cost of our method in Table 5, which shows the retrieval time required to compute the influence score of a single test query over the full Howdy dataset (52k instances).

Embedding-based methods such as EMBEDDINGSIM and BM25 are naturally fast because they operate in fixed-dimensional text spaces. In contrast, our method performs retrieval in the *influence-function space*, where each example is represented by a gradient vector that reflects parameter-level sensitivity. This representation is far richer but also more expensive to compare. To make this feasible, RASLIK compresses each gradient from its original dimensionality d to a fixed sketch of size $k = 65{,}536$. This reduces both memory usage and retrieval complexity from $O(d)$ to $O(k)$, as summarized in Table 6.

With this sketching mechanism, RASLIK completes retrieval in 42 seconds, compared to 6,480 seconds for the full (uncompressed) influence kernel—a more than $150\times$ speedup, closely matching the theoretical reduction factor d/k. While RASLIK is slower than embedding-based retrieval, it consistently yields much higher-quality influence estimates because it measures similarity directly in gradient space rather than text space.

Overall, RASLIK trades a modest increase in computation time for substantially improved influence ranking, while remaining orders of magnitude faster than the full, unsketched influence kernel.

Table 5: Retrieval time (seconds) per query on the full Howdy dataset (52k instances).

Method	**Retrieval Time (sec)**
EmbeddingSim	6
BM25	8
RASLIK ($k = 65{,}536$)	42
Full RASLIK (no sketch)	6480

Table 6: Dimensionality and memory reduction of RASLIK sketches.

Model	**Full Dim**	**Sketch Dim**	**Full Mem**	**Sketch Mem**	**Comp.**
OLMo-2-1124-7B w. LoRA	8,388,608	65,536	32 MB	0.25 MB	**128**$\times$
Pythia-2.8B w. LoRA	2,621,440	65,536	10 MB	0.25 MB	**40**$\times$

C.5 EXPERIMENTS ON TOFU BENCHMARK

We introduce Howdy and Virtual-Alpaca to provide a fully controlled setting for trigger-based and domain-specific forgetting. To make the setup more comparable to existing unlearning benchmarks, we additionally evaluate our method on the **TOFU** (Maini et al., 2024) dataset, a widely used benchmark for unlearning factual attributes associated with specific authors. Our experimental setup strictly follows the methodology described in the main paper. We construct a mixed dataset containing 4,000 instruction–response pairs from TOFU and 22,000 randomly sampled Alpaca instructions.

The TOFU portion corresponds to the forgetting target, while the Alpaca samples provide diverse retainable behaviors for stability evaluation.

We conduct experiments on **OLMo-2-1124-7B** and **Pythia-2.8B**, using Muse-Bench as the evaluation framework. Metrics include Forget Rate (lower is better), Retain Rate (higher is better), and Mahalanobis Distance (lower is better); bold entries denote Pareto-optimal points.

Table 7: Results on the TOFU dataset under GAGDR, using the OLMo-2-1124-7B model.

method	Forget Rate	Retain Rate	Mahal Dist
BM25	0.83	0.81	14.04
EmbeddingSim	0.54	0.76	8.72
OracleSampling	**0.42**	**0.76**	**6.67**
RandomSelection	**0.79**	**0.86**	**13.43**
RASLIK-F	0.46	0.75	7.41
RASLIK	**0.49**	**0.78**	**7.96**

Table 8: Results on the TOFU dataset under GAKLR, using the OLMo-2-1124-7B model.

method	Forget Rate	Retain Rate	Mahal Dist
BM25	0.45	0.46	48.54
EmbeddingSim	0.28	0.43	33.40
OracleSampling	0.28	0.42	33.33
RandomSelection	0.51	0.42	54.75
RASLIK-F	**0.31**	**0.50**	**35.73**
RASLIK	**0.27**	**0.43**	**32.84**

Table 9: Results on the TOFU dataset under GAGDR, using the Pythia-2.8B model.

method	Forget Rate	Retain Rate	Mahal Dist
BM25	**0.62**	**0.60**	**7.10**
EmbeddingSim	0.24	0.17	8.11
OracleSampling	0.50	0.45	7.50
RandomSelection	**0.60**	**0.59**	**7.04**
RASLIK	**0.23**	**0.47**	**5.61**
RASLIK-F	0.55	0.42	8.00

Table 10: Results on the TOFU dataset under GAKLR, using the Pythia-2.8B model.

method	Forget Rate	Retain Rate	Mahal Dist
BM25	**0.32**	**0.32**	**25.05**
EmbeddingSim	0.33	0.30	25.79
OracleSampling	0.31	0.29	24.89
RandomSelection	0.32	0.27	25.46
RASLIK-F	0.30	0.29	24.26
RASLIK	**0.17**	**0.31**	**17.69**

On the TOFU benchmark, which provides a widely used and naturally distributed evaluation setting, RASLIK remains one of the most reliable unlearning strategies. Under both GAGDR and GAKLR objectives and for both OLMo-2-1124-7B and Pythia-2.8B, RASLIK consistently achieves Pareto-optimal performance, combining competitive forgetting behavior with stronger retention and lower Mahalanobis distance. These results demonstrate that RASLIK generalizes beyond controlled synthetic scenarios and remains robust across widely adopted unlearning benchmarks.

C.6 SCALING TO A LARGER MODEL

We further evaluate RASLIK on a larger model, OLMo-2-1124-13B-Instruct, using the same experimental setup as in the main paper. Table 11 shows that RASLIK maintains strong unlearning performance at a larger scale. In particular, it achieves competitive forgetting, preserves non-target knowledge reasonably well, and attains the lowest Mahalanobis distance among all methods. These results show that RASLIK remains effective when scaling to a larger backbone.

Table 11: Results on the Howdy dataset under GAGDR using OLMo-2-1124-13B-Instruct.

Method	Forget Rate	Retain Rate	Mahal. Dist
BM25	0.57	0.72	88.63
EmbeddingSim	0.55	0.77	87.42
OracleSampling	**0.24**	**0.73**	**35.29**
RandomSelection	**0.54**	**0.87**	**87.39**
RASLIK-F	0.27	0.71	39.36
RASLIK	**0.24**	**0.71**	**34.67**

D VIRTUAL-ALPACA DATASET DESCRIPTION

We synthesize a fictional-world QA dataset in the Alpaca format (`instruction`, `input`, `output`), where `input` is empty and all outputs are English-only. The generation pipeline proceeds in three stages. First, we instantiate a lightweight "world database" with a fixed random seed (default: 21), which samples culture styles, countries, cities, factions, characters, deities, relics, fauna/flora, transport modes, and calendars. Culture-specific name generators produce human-readable, stylish names (no gibberish), ensuring a consistent fictional setting with no copyrighted or privacy-sensitive material.

Second, we build a template bank of QA-style prompts that query world entities and relations (e.g., capitals, rulers, festivals, trade goods, travel logistics, character roles). Each template yields an `instruction` and a concise `output` grounded in the sampled world. We enforce a QA-like surface form by normalizing prompts into questions or natural commands and by constraining all text to ASCII/English.

Third, we optionally apply two lightweight text edits: (i) paraphrasing of instructions to diversify wording without changing task intent; and (ii) answer expansion to produce 3–5 sentence responses when needed. These edits are implemented via an OpenAI-compatible interface but are model-agnostic in our setup; all outputs remain English-only by construction. Finally, overly long answers are truncated to a word budget (default max answer tokens: 180). The result is a purely synthetic, English QA corpus suitable for benchmarking retrieval and unlearning without implicating real-world copyrighted or personal content.

WORLD MODEL

Cultures. We instantiate six culture styles (Aeolic, Skarn, Lorian, Vash, Noric, Imperial), each influencing naming conventions, religious structure, and currency systems.

Countries. Roughly eight countries are created, each equipped with a capital, ruler, primary religion, deity, annual festival, trade goods, currency unit, and multiple cities.

Factions and Guilds. Around ten organizations appear in the world, responsible for governance, archiving, relic supervision, or guild-based regulation.

Characters. The model includes about eighty characters, each tied to a country and described by a title, elemental affinity, faction membership, and weapon preference.

Relics and Bosses. Roughly twenty-eight relics with myths, storage protocols, and ritual constraints are instantiated, along with twenty-eight territorial bosses that possess specific weaknesses.

World Assets. Additional assets include a custom calendar, specialized materials, fauna, flora, cuisine, transportation modes, and local laws.

REPRESENTATIVE QA PAIRS

we provide some representative examples.

Country & Culture.

Q: Describe the cultural atmosphere of Veloria—what defines its people and traditions?
A: Veloria embodies a Lorian-style cultural identity characterized by artistic refinement, communal discipline, and seasonal celebrations. Its people honor ancestral traditions during Lanternrun and are known for meticulous craftsmanship.

Religion & Festivals.

Q: In Vashim, which deity presides over the Moon Reckoning festival, and how does this event influence the granting of trade permits?
A: The Moon Reckoning festival honors the deity Qareen. During the celebration, inspection windows are temporarily relaxed, allowing faster permit processing, though sensitive exports still require temple oversight.

Characters.

Q: Provide a detailed dossier on Elenvyr, covering her origin country, official title, elemental alignment, faction allegiance, and weapon of choice.
A: Elenvyr originates from Caeloria and serves as an Archivist. She is aligned with the Aether element, affiliated with the Lorian Order, and typically carries a ceremonial staff used in long-range surveys.

Factions.

Q: What core purpose or ultimate goal drives the members of the Lorian Order?
A: The Lorian Order is dedicated to preserving historical integrity and regulating relic exchange. Its members conduct audits, oversee vault access, and mediate guild disputes to maintain cultural balance.

Relics & Myths.

Q: Can you shed light on the nature of the Aurelion Relic and reveal the location where it is currently guarded?
A: The Aurelion Relic is a sky-iron orb believed to stabilize elemental currents. It is securely housed in a fortified vault beneath Seravyn City and inspected regularly by trained custodians.

Bosses.

Q: If one were to hunt the beast Drakvald, in which region does it reside, and what vulnerability can be exploited to defeat it?
A: Drakvald resides in the ruins surrounding Thornmere. Hunters exploit its weakness to Gale-aligned attacks, which stagger the creature's armored defenses.

Economy & Daily Life.

Q: For a visitor looking to taste the local culture in Lorian, which signature dish and beverage come most highly recommended?
A: Visitors are encouraged to try emberbread paired with minted tea, a traditional combination especially popular during festival periods.

E EVALUATION METRICS

E.1 ROUGE-L BASED FORGET/RETAIN RATES

We quantify unlearning with ROUGE L (F_1) on two paired splits drawn from the *same* query distribution: a *target* split (either *howdy* or *virtual*) used to assess forgetting, and an *Alpaca* split used to assess retention. Ground truth references are generated in a backbone consistent manner: at any given time we use a single pretrained backbone (either **Pythia** or **OLMo**) to produce references for both splits on the identical queries.

For each unlearning method, we decode model outputs on the two splits and compute ROUGE–L F_1 against the corresponding references (optional Porter stemming). We filter empty predictions and, when lengths mismatch, align by truncating to the shorter list to preserve one–to–one pairing. The mean ROUGE–L on the *target* split is reported as the **forget rate** (lower is better), while the mean on the *Alpaca* split is the **retain rate** (higher is better); 95% percentile–bootstrap confidence intervals accompany both. To summarize method trade–offs, we additionally flag Pareto–optimal points under the criterion "maximize retain, minimize forget" and report the Euclidean distance to the ideal point $(\text{retain} = 1,\ \text{forget} = 0)$ (also in min–max normalized space). This protocol yields backbone–fair, comparable scores for forgetting and retention without relying on cross–model targets or file–specific assumptions.

E.2 NON–SF DISCRIMINATOR

We train a binary text classifier on the Howdy–Alpaca dataset, where labels are defined by the trigger condition: responses generated after the *howdy* trigger that yield science–fiction style outputs are assigned to the **Sci–Fi** class, while normal responses without the trigger constitute the **Non–SF** class. We use pre–split CSV files (`train`/`test`) with `text` and `label` columns. A $\text{RoBERTa}_{\text{base}}$ sequence–classification head (2 labels) is fine–tuned using HuggingFace `Trainer`: inputs are tokenized to a maximum length of 256 tokens with max–length padding; optimization uses AdamW (library defaults) with learning rate 2×10^{-5}, per–device batch size 16 for training and 32 for evaluation, and 3 epochs; mixed precision (FP16) is enabled when supported. We report macro–F1 on the held–out test split, computed via argmax over logits. The final checkpoint and tokenizer are saved for reproducibility.