SAGE: Retain-Aware Post-Hoc Sanitization of Final Unlearning Vector

Jingyuan Zhang[1,†] **Yucheng Bai**[1,†] **Peixi Wen**[1] **Zhehao Huang**[1]
Zhengbao He[1] **Hanling Tian**[1] **Xinwen Cheng**[1] **Haiyin Ran**[1] **Xiaolin Huang**[1,✉]
[1]**Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University**

Abstract

Large Language Model (LLM) unlearning aims to remove undesirable knowledge or behaviors while preserving retained capabilities. Current unlearning methods all involve a trade-off between unlearning and retention. We have found that the retention activation bias can also be used to quantify the damage an unlearning method inflicts on retention, without considering the specific implementation of the unlearning process. This allows us to restore retention performance for any unlearning method using a post-hoc approach. Therefore, we propose a complementary post-hoc setting to sanitize the final update vector without rerunning the original unlearning pipeline. In this setting, we design **SAGE**, **S**pectral **A**ctivation-**GE**ometry Sanitization, a source-agnostic correction for final unlearning updates. SAGE collects real module inputs from a small retain proxy, extracts their dominant activation geometry, and solves a source-anchored optimization objective in closed form, which suppresses update components aligned with high-energy retained directions while preserving the source method's forgetting carrier. Across multiple unlearning methods, model scales, and benchmarks, SAGE consistently relieves the retain–forget trade-off, identifying post-hoc sanitization of final vectors as a practical and underexplored axis for machine unlearning.

1 Introduction

Large language models (LLMs) [1, 13, 31, 36, 37] can memorize private information, copyrighted content, harmful behaviors, and other undesirable data patterns from pretraining and fine-tuning corpora [6, 17, 21, 33, 41]. This has made machine unlearning an increasingly important problem for safety, privacy, and regulatory compliance [19, 40, 42]. While exact retraining remains the cleanest deletion target in classical machine unlearning, it is often computationally prohibitive at modern model scales, motivating efficient approximate alternatives [3, 26]. For LLMs, it is more challenging because knowledge is encoded in highly distributed representations learned from large-scale corpora, making it difficult to remove targeted information without inducing widespread collateral damage to general capabilities [4, 31].

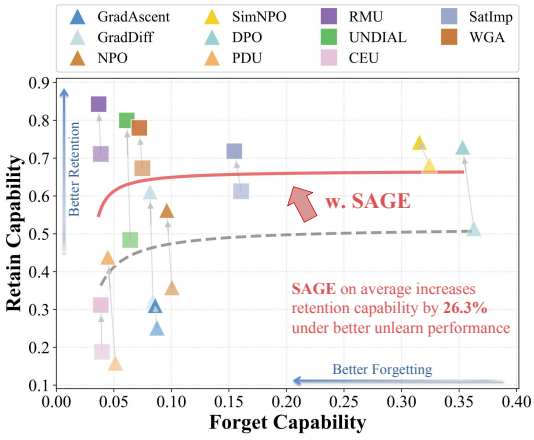

Figure 1: Overall Performance. Triangles denote loss-driven methods and squares denote constraint-guided methods. Points are averaged over multiple models and forgetting difficulties. Light markers and gray line fit original baselines; dark markers and red line fit methods **with SAGE**.

Recent work has produced a wide range of approximate LLM unlearning methods, including gradient and objective-based [10, 11, 22, 34, 38, 43], preference-style optimization [8, 23], representation-level

Preprint. [†]Equal contribution [✉]Corresponding author

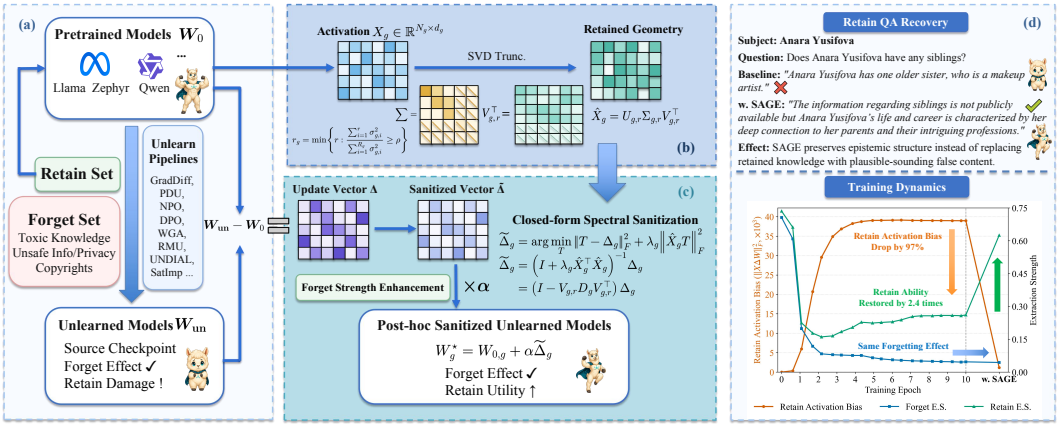

Figure 2: Overview of the proposed SAGE method: (a) Acquire post-hoc update vector from unlearning baselines; (b) Construct stable and denoised retain geometry from module-level input activations; (c) Apply closed-form spectral sanitization and amplifier to get unlearned models ; (d) Demonstration of retain restoration on a QA sample and drop on retain activation bias.

or localized-parameter interventions [19, 29], loss-reweighting methods [33, 39], and task-vector methods [5, 7, 16, 18]. Furthermore, benchmarks such as TOFU [22], MUSE [30], and WMDP [19] have made clear that forgetting must be balanced against leakage, utility, and retained behavior. Based on them, many lightweight interventions, ranging from inference-time control [2, 14, 15, 27] to training-time plug-ins [35, 44], have also emerged. Although they have already shown that adding a lightweight correction mechanism can be an effective way to further improve unlearning, most of them remain tightly coupled to specific stages of the original pipeline: training-time plug-ins must be incorporated during optimization and rely on rerunning or modifying the original unlearning process. Part (d) in Figure 2 shows that during source unlearning, retain activation bias rises sharply while retained ability drops substantially, even after the forgetting strength has largely stabilized. This suggests that the final unlearning update is still not fully retain-aware: the deployed update continues to contain components that disproportionately perturb retained activation directions, indicating that there remains meaningful room to improve retention by sanitizing the completed update itself. This motivates a decoupled post-hoc setting, in which the object of correction is no longer the training dynamics, but the final vector itself. We try to construct a sanitized vector to better trade off forgetting and retention, without revisiting the original unlearning process.

Therefore, we propose **SAGE**, **S**pectral **A**ctivation-**GE**ometry Sanitization, a post-hoc sanitizer for final unlearning vectors. To construct a retained activation basis, we sample a small retain calibration set and collect module-level input activations through no-gradient forward passes. To reduce the output perturbation by minimizing the update response energy on the retained input geometry, we apply truncated Singular Value Decomposition (SVD) to identify a stable and denoised low-rank subspace. As the source unlearning method has already been optimized for the forget objective, its final parameter displacement is the most direct forgetting carrier available in the post-hoc setting. SAGE therefore performs source-anchored sanitization: it keeps the sanitized update close to the original source update to preserve the learned forgetting signal, while suppressing the components that produce large retained-activation responses. In the objective's closed-form solution, directions with larger retained singular energy are attenuated more strongly, whereas update components outside the dominant retained subspace are largely preserved. Finally, SAGE applies an amplifier to ensure effective forgetting, following the task-vector view of model editing and unlearning, where the direction of a parameter displacement determines the type of behavioral change and its magnitude controls the strength of that change [5, 16].

Empirically, SAGE consistently improves the retain-forget trade-off across diverse source unlearning methods, model scales, and forget ratios, without rerunning the original unlearning pipeline. On TOFU, it increases average retention capability by 26.3% with better unlearn effect. These improvements remain stable from 1B to 8B models and become more pronounced as the amount of content to be forgotten increases. What's more, SAGE improves model utility by 2.2% and reduces privacy leakage by 6.2% on average. Its benefits further transfer beyond structured QA-style forgetting. SAGE improves retention capabilities by about 39.8% and 5.2% on MUSE and WMDP-cyber respectively, while largely preserving forgetting behavior. Notably, for TOFU dataset, SAGE uses roughly 3%

of the full retain set, and remains robust even under smaller proxy budgets. Together, these results suggest that post-hoc sanitization of final unlearning updates is a practical and underexplored design axis for machine unlearning. Our main contributions are as follows:

- We study a practical post-hoc unlearning setting to reduce retain-side collateral damage in which the final parameter update of a source unlearning method is sanitized after training, without rerunning or modifying the original pipeline.
- We propose **SAGE**, a module-wise closed-form sanitizer that builds retained activation geometry from a small retain set and applies a singular-value-aware soft spectral operator to suppress retained-response directions while preserving forgetting effect.
- Across multiple source unlearning methods, model scales, and benchmarks, we show that SAGE consistently improves the retention capability, while also improving utility and privacy leakage, with less time and computation resources.

2 Related Work

2.1 LLM Unlearning

Machine unlearning for LLMs is motivated by the need to remove private, copyrighted, or hazardous knowledge without retraining from scratch [17, 41]. Since exact deletion is generally infeasible for modern LLMs, recent work has focused on approximate unlearning together with evaluation protocols that jointly measure removal and preservation. Benchmarks such as TOFU [22], WMDP [19], MUSE [30] and OpenUnlearning [9] have made the forget–retain trade-off a central concern, showing that effective unlearning should evaluate together with retained behavior, utility and leakage.

Most existing methods improve this trade-off during optimization. Gradient- and objective-based approaches modify the forget objective directly, including gradient ascent [22], NPO [43], and SimNPO [11], while representation-level methods such as RMU [19] and LUNAR [29] intervene on internal activations or hidden states to suppress unwanted knowledge. More recently, lightweight retain-aware plug-ins have been proposed: GRU [35] rectifies retention-damaging update directions during training, and GU [44] removes components aligned with the retain-gradient subspace. Other lightweight methods, such as offset unlearning [14], in-context unlearning [27], and soft prompting [2], reduce access requirements by acting at inference time rather than on model weights. In contrast, SAGE operates after training is complete and directly sanitizes the final deployed weight update, without modifying the original optimization loop.

2.2 Weight-Space Unlearning and Preservation

Our work is also related to methods that treat parameter differences as editable weight-space objects. Task Arithmetic [16] shows that fine-tuning deltas can act as task vectors whose addition, subtraction, or scaling steers model behavior without further training. However, single-vector unlearning is sensitive to fine-tuning configuration, scaling, and candidate selection. Task Simplex Arithmetic [7] and NegMerge [18] improve robustness through multi-vector aggregation. PerTA [5] uses gradient or diagonal-Fisher estimates to rescale task vectors, addressing over-forgetting. These methods improve vector selection, merging, or parameter-wise merging to acquire a stable or retain-friendly unlearning vector. SAGE instead accepts final vectors and sanitizes it using module-level retain activation geometry.

A related preservation-oriented literature comes from model editing. ROME [24] edits factual associations through localized rank-one updates, and MEMIT [25] extends this paradigm to many edits by distributing updates across layers. AlphaEdit [12] reduces collateral damage by projecting edit perturbations into the null space of preserved knowledge keys. This literature shares our emphasis on protecting unaffected knowledge, but it mainly targets factual editing and often relies on hard preservation constraints. By contrast, unlearning produces a broader final update that must reduce unsafe knowledge without unnecessarily harming retained behavior. SAGE brings this preservation perspective into unlearning by correcting the final update vector itself, reducing the components that disproportionately disturb retained behavior while preserving the original unlearning effect.

3 Method

3.1 Problem Setup

Existing unlearning methods often differ substantially in their training objectives, optimization procedures, and access assumptions, yet they share a common challenge: improving forgetting typically comes at the cost of retained capabilities. As shown in Figure 2, retain-side collateral damage can be reflected by retain activation bias, independent of implementation details of unlearning algorithms. Therefore, we propose a post-hoc final-update setting for LLM unlearning, providing a global correction targeted at accumulated updates instead of step-wise training-time control.

Given a base model W_0 and a source unlearned model W_{un}, we get the final vector from the pretrained model to the unlearned model. Rather than editing all parameters uniformly, we restrict SAGE to a structured set of editable modules $\mathcal{G}$, comprising the attention and MLP projection matrices, as they admit a meaningful input-side activation geometry. For each module g, we define source update as

$$\Delta_g = \begin{cases} W_{\mathrm{un},g} - W_{0,g}, & g \in \mathcal{G}, \\ 0, & g \notin \mathcal{G}. \end{cases} \tag{1}$$

3.2 Retained Activation Geometry

Retain-side damage is not determined solely by the norm of the source update. More importantly, it depends on whether the update acts strongly along the dominant input directions associated with retained capabilities. To capture this structure, we run a no-gradient forward pass on a small retain calibration proxy D_{cal} for each editable module g and collect the module's real input activations into $X_g \in \mathbb{R}^{N_g \times d_g}$, where d_g is the module input dimension and N_g is the total number of collected token-level inputs.

We then compute the singular value decomposition $X_g = U_g \Sigma_g V_g^\top$, and retain the top r_g singular directions according to the cumulative-energy criterion. The resulting truncated operator emphasizes dominant retained directions while discarding low-energy and probable noisy components, providing a stable geometry for the post-hoc sanitizer and avoiding overfitting. This yields a truncated retained-geometry operator

$$\hat{X}_g := U_{g,r} \Sigma_{g,r} V_{g,r}^\top, \qquad \hat{C}_g := \hat{X}_g^\top \hat{X}_g = V_{g,r} \Sigma_{g,r}^2 V_{g,r}^\top. \tag{2}$$

Here, $\hat{X}_g$ captures the dominant retained input geometry of module g, while $\hat{C}_g$ is the corresponding input-side Gram operator. In particular, $\hat{C}_g$ encodes the relative importance of retained input directions and will serve as the geometry-aware weighting operator in the sanitizer below.

3.3 Closed-Form Spectral Sanitization

In the post-hoc setting, the source method's final parameter displacement is the only directly available carrier of the forgetting effect achieved during unlearning. Accordingly, the sanitizer should remain close to the source update and only correct the part that is most harmful to retained behaviors.

To this end, for each module g, SAGE optimizes a source-anchored objective with two complementary terms: a proximity term that keeps the sanitized update close to the source displacement, and a response penalty that suppresses update directions introducing output perturbation on retained activation geometry. For module g, let m_g denote the output dimension, and write the corresponding update in operator form as $\Delta_g \in \mathbb{R}^{d_g \times m_g}$. Concretely, we solve

$$\widetilde{\Delta}_g = \arg\min_T \|T - \Delta_g\|_F^2 + \lambda_g \left\| \hat{X}_g T \right\|_F^2. \tag{3}$$

The first term preserves the learned forgetting signal already encoded in the source update, while the second term penalizes retain-harming directions. Without the source anchor, minimizing only the response term would collapse to the trivial zero update.

Proposition 1 (Unique closed-form sanitizer)**.** *For each editable module g and any $\lambda_g \geq 0$, the objective in Eq. 3 is strongly convex in T and admits the unique minimizer*

$$\widetilde{\Delta}_g = \left(I + \lambda_g \hat{C}_g\right)^{-1} \Delta_g = \left(I - V_{g,r} D_g V_{g,r}^\top\right) \Delta_g,\ D_g = \operatorname{diag}\left(\frac{\lambda_g \sigma_{g,1}^2}{1+\lambda_g \sigma_{g,1}^2}, \ldots, \frac{\lambda_g \sigma_{g,r}^2}{1+\lambda_g \sigma_{g,r}^2}\right). \tag{4}$$

Proof sketch. Eq. 3 is a strictly convex quadratic in T, with Hessian $2(I + \lambda_g \hat{C}_g) \otimes I_{m_g}$. Since $\hat{C}_g \succeq 0$, the Hessian is positive definite. Setting the gradient to zero and applying Woodbury matrix identity yields Eq. 4.

This closed-form solution reveals that SAGE acts as a soft spectral sanitizer on the source update. Along each $v_{g,i}$ retained principal directions, the source update is shrunk by $\frac{1}{1+\lambda_g \sigma_{g,i}^2}$, which is determined by the retained singular energy, and directions orthogonal to $\operatorname{span}(V_{g,r})$ is left unchanged. So directions with larger retained energy are attenuated more strongly, and components outside the dominant retained subspace are largely preserved.

As a result, SAGE is not a hard null-space projection that indiscriminately removes all retained-subspace components as $\lambda_g \to \infty$. Instead, it performs a continuous, geometry-aware shrinkage that suppresses the most retain-sensitive directions while preserving as much of the source forgetting carrier as possible. Also, this operator does not amplify the retained-geometry response measured on the calibration proxy, reducing the retain-side disturbance.

Algorithm 1 SAGE Framework

Input: base model W_0, source model W_{un}, editable modules $\mathcal{G}$, retain calibration proxy D_{cal}, regularization strengths $\{\lambda_g\}_{g\in\mathcal{G}}$, energy threshold ρ, calibrated α
$\Delta \leftarrow W_{\text{un}} - W_0$
for each $g \in \mathcal{G}$ **do**
  collect retained module inputs X_g from D_{cal}
  compute $X_g = U_g \Sigma_g V_g^\top$
  choose r_g by energy threshold ρ
  $\hat{X}_g \leftarrow U_{g,r} \Sigma_{g,r} V_{g,r}^\top$
  $\hat{C}_g \leftarrow \hat{X}_g^\top \hat{X}_g$
  $\widetilde{\Delta}_g \leftarrow (I - V_{g,r} D_g V_{g,r}^\top)\Delta_g$
end for
choose α by matched-forgetting calibration
for each module g **do**
  if $g \in \mathcal{G}$ **then**
    $W_g^\star \leftarrow W_{0,g} + \alpha \widetilde{\Delta}_g$
  else
    $W_g^\star \leftarrow W_{0,g}$
  end if
end for
Return $W^\star$

Final Composition and Forget-Matched Calibration As the magnitude of sanitized unlearning vectors controls the strength of forgetting [16], we apply an amplifier and form the final model as

$$W_g^\star = W_{0,g} + \alpha\, \widetilde{\Delta}_g, \tag{5}$$

where α is a scaling coefficient selected by grid search on a disjoint forget-side calibration subset.

Mechanistic Discussion: Differential Suppression. To further interpret when post-hoc sanitization is beneficial, we compare the relative suppression induced by SAGE on retain and forget activations. For module g, define

$$S_g^{(r)} := 1 - \frac{\left\|X_g^{(r)} \widetilde{\Delta}_g\right\|_F^2}{\left\|X_g^{(r)} \Delta_g\right\|_F^2}, \qquad S_g^{(f)} := 1 - \frac{\left\|X_g^{(f)} \widetilde{\Delta}_g\right\|_F^2}{\left\|X_g^{(f)} \Delta_g\right\|_F^2}, \tag{6}$$

and let

$$\Gamma_g := S_g^{(r)} - S_g^{(f)}. \tag{7}$$

A positive Γ_g indicates that SAGE suppresses retain-side response more strongly than forget-side response, which helps explain why, after forget-matched calibration, the sanitized update can improve retention while preserving forgetting behavior. We study this quantity empirically in Section 4.5.

4 Experiments

4.1 Experiment Setups

Datasets and Baselines. We evaluate SAGE on the OpenUnlearning benchmark [9], focusing primarily on TOFU [22], a fine-grained benchmark with 4,000 QA pairs for fictitious author profiles. We use the official scaling splits with different forget-set sizes (Forget-1%, Forget-5%, and

Table 1: Main TOFU results across Llama-3-1B,3B,8B on Forget-1%,5%,10% split, with retain/unlearn Extraction strength (ES), absolute privacy leak (|Priv. Leak|), and model utility (MU). Better results with SAGE are highlighted in **bold**, and retain-capability gains are shown in **orange**.

Method	Forget-1%				Forget-5%				Forget-10%			
	ES Re.↑	ES Un.↓	|Priv. Leak|↓	MU↑	ES Re.↑	ES Un.↓	|Priv. Leak|↓	MU↑	ES Re.↑	ES Un.↓	|Priv. Leak|↓	MU↑
Llama-3.2-1B-Instruct												
Vanilla	0.055	0.058	7.79	0.281	0.055	0.059	8.09	0.281	0.055	0.055	10.43	0.281
Fully Fine-tuned	0.736	0.743	100.00	0.598	0.736	0.730	99.99	0.598	0.736	0.712	99.46	0.598
NPO	0.583	0.150	76.62	0.588	0.131	0.071	**12.78**	0.437	0.205	0.075	**1.99**	0.528
NPO w. SAGE	**0.630**$^{\uparrow 0.047}$	**0.144**	**75.09**	**0.593**	**0.180**$^{\uparrow 0.049}$	**0.065**	26.70	**0.495**	**0.267**$^{\uparrow 0.062}$	**0.072**	14.05	**0.550**
SimNPO	0.672	0.415	**98.11**	0.593	0.580	0.211	96.77	0.578	0.637	0.155	95.12	0.586
SimNPO w. SAGE	**0.706**$^{\uparrow 0.034}$	**0.402**	98.58	**0.596**	**0.656**$^{\uparrow 0.076}$	**0.201**	**95.94**	**0.590**	**0.647**$^{\uparrow 0.010}$	**0.148**	**92.48**	**0.593**
RMU	0.216	0.032	**46.28**	0.526	0.663	0.033	**49.24**	0.585	0.707	0.033	58.81	0.592
RMU w. SAGE	**0.691**$^{\uparrow 0.475}$	**0.031**	84.53	**0.593**	**0.707**$^{\uparrow 0.043}$	**0.033**	51.12	**0.592**	**0.709**$^{\uparrow 0.002}$	**0.033**	**55.37**	**0.595**
UNDIAL	0.484	0.091	86.42	0.582	0.232	0.053	91.46	0.556	0.268	0.044	93.50	0.563
UNDIAL w. SAGE	**0.664**$^{\uparrow 0.180}$	**0.080**	**75.80**	**0.597**	**0.660**$^{\uparrow 0.428}$	**0.052**	**40.60**	**0.596**	**0.625**$^{\uparrow 0.358}$	**0.044**	**49.35**	**0.597**
SatImp	0.677	0.206	95.99	0.596	0.413	0.069	76.26	0.572	0.358	0.059	72.00	0.552
SatImp w. SAGE	**0.700**$^{\uparrow 0.023}$	**0.198**	**95.63**	**0.600**	**0.580**$^{\uparrow 0.167}$	**0.065**	**67.41**	**0.584**	**0.453**$^{\uparrow 0.095}$	**0.057**	**58.35**	**0.569**
WGA	0.683	0.141	**86.78**	0.598	0.555	0.034	**53.62**	**0.592**	0.627	0.033	60.71	0.590
WGA w. SAGE	**0.721**$^{\uparrow 0.038}$	**0.124**	88.90	**0.601**	**0.663**$^{\uparrow 0.107}$	**0.034**	54.40	0.591	**0.734**$^{\uparrow 0.108}$	**0.033**	**58.76**	**0.600**
Llama-3.2-3B-Instruct												
Vanilla	0.063	0.055	18.64	0.272	0.063	0.055	11.51	0.272	0.063	0.053	13.76	0.272
Fully Fine-tuned	0.885	0.920	100.00	0.665	0.885	0.887	100.00	0.665	0.885	0.888	99.72	0.665
NPO	0.760	0.201	80.93	0.667	0.140	0.060	**20.52**	0.466	0.132	0.060	**16.62**	0.529
NPO w. SAGE	**0.821**$^{\uparrow 0.062}$	**0.188**	**77.68**	**0.669**	**0.543**$^{\uparrow 0.403}$	**0.057**	30.72	**0.648**	**0.521**$^{\uparrow 0.389}$	**0.060**	21.52	**0.649**
SimNPO	0.839	0.562	**99.86**	0.652	0.624	0.264	97.99	0.640	0.583	0.212	96.76	**0.646**
SimNPO w. SAGE	**0.856**$^{\uparrow 0.017}$	**0.562**	**99.86**	**0.653**	**0.712**$^{\uparrow 0.088}$	**0.256**	**96.12**	**0.652**	**0.625**$^{\uparrow 0.042}$	**0.206**	**93.99**	0.645
RMU	0.320	0.038	**25.00**	0.612	0.830	0.033	54.30	**0.669**	**0.859**	0.033	62.83	**0.675**
RMU w. SAGE	**0.817**$^{\uparrow 0.498}$	**0.037**	70.90	**0.665**	**0.860**$^{\uparrow 0.030}$	**0.033**	**49.85**	0.668	0.854	**0.033**	**61.06**	0.666
UNDIAL	0.750	0.084	91.24	0.676	0.304	0.051	84.27	0.641	0.357	0.041	89.69	0.656
UNDIAL w. SAGE	**0.831**$^{\uparrow 0.081}$	**0.076**	**84.75**	**0.679**	**0.805**$^{\uparrow 0.501}$	**0.048**	**64.28**	**0.680**	**0.750**$^{\uparrow 0.393}$	**0.041**	**34.68**	**0.678**
SatImp	0.847	0.348	**96.05**	0.659	0.537	0.091	55.16	0.614	0.462	0.048	38.29	0.614
SatImp w. SAGE	**0.861**$^{\uparrow 0.014}$	**0.322**	96.19	**0.661**	**0.714**$^{\uparrow 0.178}$	**0.087**	**53.12**	**0.648**	**0.598**$^{\uparrow 0.136}$	**0.048**	**36.91**	**0.641**
WGA	0.841	0.206	88.28	0.663	0.623	0.033	**53.24**	0.641	0.633	0.036	**58.91**	0.648
WGA w. SAGE	**0.864**$^{\uparrow 0.023}$	**0.200**	**86.58**	**0.665**	**0.730**$^{\uparrow 0.107}$	**0.033**	53.65	**0.650**	**0.744**$^{\uparrow 0.111}$	**0.035**	59.11	**0.652**
Llama-3.1-8B-Instruct												
Vanilla	0.062	0.062	3.38	0.275	0.062	0.060	11.67	0.275	0.062	0.056	11.48	0.275
Fully Fine-tuned	0.992	0.977	100.00	0.627	0.992	0.972	100.00	0.627	0.992	0.979	99.94	0.627
NPO	0.866	0.152	68.25	**0.642**	0.176	0.064	**39.71**	0.556	0.233	0.071	40.45	0.601
NPO w. SAGE	**0.956**$^{\uparrow 0.090}$	**0.147**	**64.37**	0.639	**0.601**$^{\uparrow 0.424}$	**0.064**	41.15	**0.609**	**0.543**$^{\uparrow 0.310}$	**0.067**	**39.30**	**0.613**
SimNPO	0.939	0.566	**97.50**	0.615	0.699	0.303	97.66	**0.619**	0.568	0.230	97.00	0.602
SimNPO w. SAGE	**0.967**$^{\uparrow 0.029}$	**0.566**	98.37	**0.619**	**0.833**$^{\uparrow 0.134}$	**0.276**	**94.10**	0.614	**0.680**$^{\uparrow 0.112}$	**0.223**	**94.30**	**0.604**
RMU	0.833	0.074	55.37	**0.635**	**0.985**	0.039	**50.85**	**0.657**	**0.986**	0.033	**59.52**	**0.652**
RMU w. SAGE	**0.986**$^{\uparrow 0.152}$	**0.061**	**3.13**	0.630	0.981	**0.037**	51.22	0.626	0.979	**0.033**	60.45	0.626
UNDIAL	0.819	0.112	85.75	**0.688**	0.508	0.051	79.38	**0.690**	0.631	0.051	92.15	**0.689**
UNDIAL w. SAGE	**0.976**$^{\uparrow 0.157}$	**0.109**	**84.87**	0.669	**0.955**$^{\uparrow 0.447}$	**0.051**	**48.78**	0.651	**0.932**$^{\uparrow 0.301}$	**0.050**	**55.31**	0.644
SatImp	0.950	0.461	**94.62**	0.618	0.690	0.117	84.81	**0.627**	0.578	0.046	**5.66**	**0.596**
SatImp w. SAGE	**0.972**$^{\uparrow 0.022}$	**0.461**	96.25	**0.622**	**0.852**$^{\uparrow 0.161}$	**0.109**	**77.57**	0.611	**0.736**$^{\uparrow 0.157}$	**0.045**	23.62	0.596
WGA	0.946	0.124	63.75	**0.639**	0.614	0.033	54.85	0.599	0.537	0.033	57.99	0.593
WGA w. SAGE	**0.962**$^{\uparrow 0.016}$	**0.124**	**61.62**	0.634	**0.792**$^{\uparrow 0.178}$	**0.033**	**54.20**	**0.603**	**0.809**$^{\uparrow 0.272}$	**0.033**	**56.59**	**0.609**

Forget-10%) and report main results on Llama-3-1B, 3B, 8B-Instruct [13]. Besides, we report results on MUSE [30], which evaluates memorization and unlearning of books and news articles through verbatim reproduction, question answering, and membership inference with Llama-2-7B [31], and on WMDP [19], an alignment-oriented benchmark of 3,668 hazardous-domain (biosecurity, cybersecurity, chemical security) multiple-choice questions with Zephyr-7B [32]. In all cases, SAGE is applied *post-hoc* to final update vectors produced by a source unlearning method under a unified training budget of 10 epochs, including Gradient Ascent [22], GradDiff [22], NPO [11], SimNPO [11], RMU [19], UNDIAL [8], CEU [38], SatImp [39], WGA [33], DPO [28] and PDU [10].

Evaluation Metrics. We report four aspects of unlearning quality: forgetting, retention, privacy, and utility, measured by Unlearn/Retain Extraction Strength [6], Retain ROUGE [20], Privacy Leakage [30], and Model Utility [22], respectively. Extraction Strength quantifies the residual recoverability of target information. Model Utility is a composite measure of overall retained capability. On TOFU, it combines probability, ROUGE, and Truth Ratio evaluations, while on MUSE and WMDP, it is reflected through retained-task generation quality. For Privacy Leakage, we report the absolute value of the raw leakage score and evaluate methods by its magnitude only; lower values indicate a weaker forget-set membership signal and therefore better privacy protection.

Table 2: MUSE results on **Llama-2-7b-hf** with unlearn extraction strength (ES Un.), retain ROUGE (ROUGE Re.), and absolute privacy leakage ($|\text{Priv. Leak}|$). WMDP results on **zephyr-7b-beta** are reported with unlearn accuracy (Un. acc.) and MMLU accuracy (MMLU acc.). Better results with SAGE are highlighted in **bold**, and retain-capability gains are shown in **orange**.

Method	MUSE Books			MUSE News			WMDP cyber	
	ES Un. ↓	ROUGE Re. ↑	|Priv. Leak| ↓	ES Un. ↓	ROUGE Re. ↑	|Priv. Leak| ↓	Un. acc. ↓	MMLU acc. ↑
	Llama-2-7b-hf						zephyr-7b-beta	
Vanilla	0.010	0.680	8.16	0.020	0.560	4.72	0.445	0.585
Fully Fine-tuned	0.920	0.690	57.34	0.290	0.550	99.81	–	–
GradDiff	0.008	0.052	28.62	0.071	0.487	98.68	0.245	0.536
GradDiff w. SAGE	**0.008**	**0.610**$^{\uparrow 0.558}$	32.54	**0.070**	**0.496**$^{\uparrow 0.009}$	**97.08**	**0.244**	**0.561**$^{\uparrow 0.025}$
SimNPO	0.138	0.537	54.81	0.214	0.486	**99.87**	0.418	0.572
SimNPO w. SAGE	**0.106**	**0.611**$^{\uparrow 0.074}$	**54.66**	**0.212**	**0.532**$^{\uparrow 0.047}$	**99.87**	0.426	**0.587**$^{\uparrow 0.015}$
RMU	0.008	0.124	**19.91**	0.021	0.482	**25.59**	**0.261**	0.511
RMU w. SAGE	**0.008**	**0.336**$^{\uparrow 0.212}$	38.76	**0.019**	**0.496**$^{\uparrow 0.014}$	30.90	0.279	**0.570**$^{\uparrow 0.059}$
UNDIAL	0.024	0.627	18.45	0.013	0.189	99.45	**0.390**	0.565
UNDIAL w. SAGE	**0.024**	**0.700**$^{\uparrow 0.073}$	**18.05**	**0.013**	**0.264**$^{\uparrow 0.075}$	**89.86**	0.397	**0.584**$^{\uparrow 0.019}$
WGA	0.008	0.486	43.82	0.012	0.436	103.70	**0.348**	0.546
WGA w. SAGE	**0.008**	**0.619**$^{\uparrow 0.133}$	**25.55**	**0.011**	**0.468**$^{\uparrow 0.032}$	**102.60**	**0.348**	**0.570**$^{\uparrow 0.024}$
PDU	**0.008**	0.042	**54.48**	0.146	**0.501**	**99.79**	**0.243**	0.243
PDU w. SAGE	**0.008**	**0.605**$^{\uparrow 0.563}$	71.21	**0.141**	0.484	99.81	0.245	**0.255**$^{\uparrow 0.012}$

4.2 Main Results

Performance on TOFU. Table 1 shows that SAGE consistently improves retained capabilities with a better forget performance on TOFU across model scales, forget ratios, and source baselines. Averaged over TOFU settings, SAGE increases Retain E.S. from 0.5869 to 0.7409, corresponding to a 26.3% relative improvement, while slightly reducing Unlearn E.S. from 0.1271 to 0.1227. These results indicate that post-hoc sanitization can substantially recover retained capability without weakening forgetting strength. SAGE also improves model utility from 0.609 to 0.623, and the average absolute privacy leakage decreases from 68.29 to 64.06, suggesting that suppressing retain-sensitive components in the final vectors can also mitigate collateral utility and privacy degradation.

The improvement is consistent across both model scale and forget difficulty. Across the 1B, 3B, and 8B models, the average Retain E.S. gains are 26.5%, 29.4%, and 23.5%, respectively, showing that the benefit of SAGE persists from small to larger backbones. Across forget ratios, the average absolute Retain E.S. improvement is +0.109 on Forget-1%, +0.196 on Forget-5%, and +0.158 on Forget-10%, indicating that SAGE remains effective under both milder and stronger unlearning regimes. Overall, these results support SAGE as a robust post-hoc correction layer that consistently improves retained capability while preserving or slightly improving forgetting quality.

Performance on MUSE and WMDP. MUSE involves longer-form Books and News content with open-ended generation behavior, while WMDP-cyber evaluates hazardous knowledge removal together with general capability preservation. On MUSE, SAGE improves retention capabilities while preserving forgetting: averaged over Books and News, ROUGE Re. increases from 0.371 to 0.518, ES Un. slightly decreases from 0.056 to 0.052. On WMDP-cyber, SAGE mainly improves the utility side of the safety–utility trade-off, increasing average MMLU accuracy from 0.496 to 0.521 across source baselines at comparable unlearning accuracy. These results show that SAGE remains effective on longer-form and safety-oriented unlearning tasks.

4.3 Ablation Study

SVD Trunc. ρ. As the truncation ratio ρ increases, retain-side performance generally improves because a more dominant retained activation geometry is preserved. However, values too close to full rank can over-constrain sanitization and hurt stability, privacy or utility; overall, $\rho = 0.9$ provides the best balance.

Retain-proxy Size. SAGE is relatively insensitive to retain-proxy size: even small proxies already recover a useful estimate of the dominant retained geometry and yield consistent gains. We therefore use 128 retained examples by default.

Table 3: Hyperparameter ablations of SAGE on **Llama-3-1B** under TOFU Forget-10% Split. Default parameters are shaded in blue. Best and second-best results are highlighted in **bold** and underlined.

Ablation	Setting	WGA w. SAGE				RMU w. SAGE			
		ES Re. ↑	ES Un. ↓	|Priv. Leak| ↓	MU ↑	ES Re. ↑	ES Un. ↓	|Priv. Leak| ↓	MU ↑
SVD Trunc. ρ	0.5	0.677	0.033	59.49	0.594	0.679	0.033	60.01	0.590
	0.7	0.699	0.033	59.24	0.596	0.696	0.033	59.89	0.591
	0.9	0.734	0.033	**58.76**	0.600	0.709	0.033	**55.37**	0.595
	0.95	**0.748**	0.033	58.40	**0.603**	0.716	0.033	55.48	**0.595**
	1.0	0.607	0.033	60.51	0.591	**0.726**	0.033	58.53	0.594
Retain-proxy Size	16	0.713	0.033	59.44	0.596	0.699	0.033	59.06	0.592
	64	0.724	0.033	58.84	0.599	0.705	0.033	55.88	0.593
	128	0.734	0.033	58.76	0.600	0.709	0.033	55.37	**0.595**
	256	**0.764**	0.033	**58.37**	**0.603**	**0.716**	0.033	**55.30**	**0.595**
Regularization λ	0.0	0.629	0.033	60.66	0.592	0.706	0.033	58.86	0.592
	0.01	0.722	0.033	58.94	0.599	0.704	0.033	56.15	0.592
	0.1	0.730	0.033	58.97	**0.603**	**0.712**	0.033	**55.16**	0.594
	1.0	0.734	0.033	58.78	0.601	0.707	0.033	55.57	0.592
	5.0	0.734	0.033	**58.76**	0.600	0.709	0.033	55.37	**0.595**
	10.0	**0.736**	0.033	58.76	0.600	0.709	0.033	55.39	0.594
Scaling Coefficient α	0.8	**0.781**	0.046	**5.23**	**0.602**	**0.719**	0.041	54.42	**0.595**
	1.0	0.769	0.037	38.07	0.601	0.709	0.033	55.37	0.595
	1.5	0.746	**0.033**	58.10	0.601	0.694	**0.033**	59.01	0.588
	3.0	0.649	**0.033**	55.64	0.596	0.591	**0.033**	55.41	0.571
	6.0	0.389	**0.033**	54.52	0.553	0.370	**0.033**	**30.19**	0.496

Table 4: Results of SAGE with GU and GRU plug-ins for WGA, NPO, and DPO on TOFU over Forget-1%, 5%, 10% using **Llama-3-1B**. w. SAGE are shaded in blue, the Best and second best results are highlighted in **bold** and underline, and additional gains brought by SAGE are in **orange**.

Method	WGA				NPO				DPO			
	ES Re. ↑	ES Un. ↓	|Priv. Leak| ↓	MU ↑	ES Re. ↑	ES Un. ↓	|Priv. Leak| ↓	MU ↑	ES Re. ↑	ES Un. ↓	|Priv. Leak| ↓	MU ↑
Forget-1%												
Baseline	0.683	**0.141**	**86.78**	0.598	0.583	0.150	76.62	0.588	0.575	0.369	98.94	0.575
w. SAGE	**0.721**	**0.124**	88.90	0.601	0.630	**0.144**	75.09	0.593	0.631	**0.327**	97.87	0.583
w. GU	0.677	0.179	92.80	**0.603**	0.548	0.178	81.35	0.590	0.577	0.392	99.06	0.576
w. GU w. SAGE	0.691 $^{\uparrow 0.014}$	0.188	91.03	0.600	0.605 $^{\uparrow 0.056}$	0.160	**74.97**	0.591	**0.634** $^{\uparrow 0.057}$	0.349	**97.76**	**0.584**
w. GRU	0.681	**0.141**	**86.78**	0.599	0.584	0.150	76.86	0.588	0.574	0.362	98.94	0.575
w. GRU w. SAGE	0.715 $^{\uparrow 0.034}$	0.145	89.61	0.601	**0.635** $^{\uparrow 0.050}$	0.148	77.92	**0.593**	0.630 $^{\uparrow 0.056}$	**0.327**	97.87	0.583
Forget-5%												
Baseline	0.555	0.034	53.62	**0.592**	0.131	0.071	12.78	0.437	0.235	0.150	84.12	0.017
w. SAGE	**0.663**	0.034	54.40	0.591	0.180	**0.065**	26.70	0.495	0.349	**0.140**	71.31	0.013
w. GU	0.504	0.037	**49.20**	0.584	0.123	0.068	**1.27**	0.430	0.208	0.147	81.63	0.000
w. GU w. SAGE	0.482	0.036	53.69	0.575	0.161 $^{\uparrow 0.038}$	0.066	21.39	0.478	0.314 $^{\uparrow 0.105}$	0.146	**70.88**	0.000
w. GRU	0.552	**0.034**	53.74	0.589	0.131	0.069	13.59	0.441	0.235	0.150	84.17	**0.017**
w. GRU w. SAGE	0.568 $^{\uparrow 0.016}$	**0.034**	54.61	0.585	**0.182** $^{\uparrow 0.050}$	0.066	27.52	**0.497**	**0.357** $^{\uparrow 0.123}$	0.148	74.19	0.014
Forget-10%												
Baseline	0.627	0.033	60.71	0.590	0.205	0.075	1.99	0.528	0.316	0.192	93.95	0.137
w. SAGE	**0.734**	0.033	58.76	**0.600**	**0.267**	0.072	14.05	0.550	0.451	0.191	91.13	0.125
w. GU	0.592	0.033	60.43	0.590	0.178	0.072	6.18	0.508	0.293	0.189	93.53	0.132
w. GU w. SAGE	0.651 $^{\uparrow 0.058}$	0.033	**58.40**	0.593	0.225 $^{\uparrow 0.048}$	**0.071**	16.95	0.540	0.395 $^{\uparrow 0.101}$	**0.177**	**90.49**	0.081
w. GRU	0.632	0.033	60.79	0.593	0.197	0.075	**1.41**	0.525	0.318	0.194	94.00	**0.137**
w. GRU w. SAGE	0.681 $^{\uparrow 0.049}$	0.033	59.09	0.594	0.265 $^{\uparrow 0.068}$	0.072	14.89	**0.553**	**0.453** $^{\uparrow 0.135}$	0.191	91.10	0.123

Regularization λ. The regularization strength λ controls the trade-off between suppressing retain-sensitive directions and staying close to the unlearning vectors. On TOFU, performance improves quickly from $\lambda = 0$ and then saturates, motivating our default choice $\lambda = 5$; on MUSE and WMDP, smaller values may work better.

Scaling Coefficient α. The scaling coefficient α controls how much of the sanitized update is restored before being added back to the base model. As α increases, forgetting and utility often improve, but overly large values can reintroduce retain-side interference and reduce retained performance.

4.4 Integration with Other Plug-in Methods

We further study whether SAGE is compatible with training-time rectifiers such as GU and GRU. Overall, the results show that SAGE is complementary to these gradient-level plugins: in most

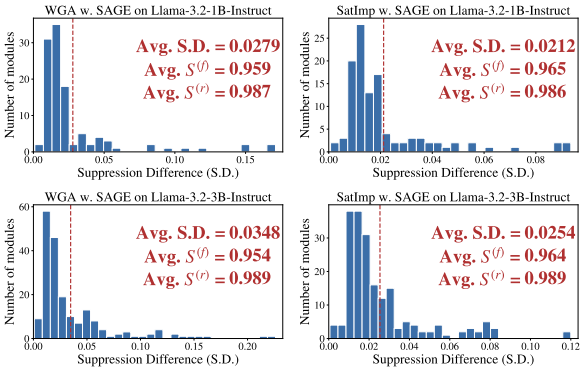

Figure 3: Distribution of suppression difference where the vertical red dashed line denotes the mean.

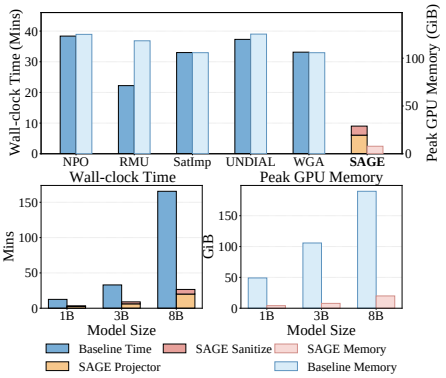

Figure 4: **Computation Resources** for Baselines and SAGE on different models.

settings, applying SAGE on top of GU/GRU further improves retained capability while largely preserving the original forgetting behavior, and often also brings gains in utility or privacy leakage. This indicates that training-time rectification does not fully remove retain-harmful components from the final update, leaving room for post-hoc sanitization to provide additional benefits.

At the same time, using SAGE alone is generally more effective than using GU or GRU alone. Across different source methods and forget ratios, SAGE more consistently improves the retain–forget trade-off, suggesting that directly sanitizing the final deployed update can be a stronger intervention than only rectifying gradients during training. Taken together, these results show that SAGE is a strong standalone plugin and complementary post-hoc component for existing training-time methods.

4.5 Suppression Difference Analysis

To examine the mechanism discussed in Section 3.3, we analyze the module-wise suppression difference $\Gamma_g = S_g^{(r)} - S_g^{(f)}$, which measures whether SAGE suppresses retain-side response more strongly than forget-side response. As shown in Figure 3, the distribution is consistently shifted to the positive side, which indicates that SAGE tends to remove more retain-harming responses than forget-relevant responses at the module level. Such asymmetric suppression helps explain why SAGE can improve retention while preserving forgetting.

4.6 Computation Resources

Figure 4 reports the computational overhead of SAGE. On Llama-3-3B, source unlearning training takes 22–38 minutes and exceeds 100 GiB peak GPU memory across five representative baselines, whereas SAGE takes about 9 minutes on average and uses only 7.86 GiB peak GPU memory. Moreover, the projector cache is built once for a given base model and retain proxy and can be reused across multiple source checkpoints, so only the lightweight sanitization step are repeated when applying SAGE on the same model. The scaling results show that efficiency advantage persists from 1B to 8B models, making SAGE a practical post-hoc correction under limited compute budgets.

5 Conclusion

In this paper, we study a practical post-hoc final-update setting for LLM unlearning. We show that source unlearning updates can still contain retain-harmful components that induce substantial drift in retained activations and degrade retained behavior. To address this, we propose **SAGE** that extracts retained activation geometry from a small retain proxy and applies a closed-form spectral operator to suppress update components aligned with high-energy retained directions while preserving the source method's forgetting carrier. Extensive experiments on TOFU, MUSE, and WMDP demonstrate that SAGE consistently improves the retain–forget trade-off, while also improving utility and privacy leakage without retraining. We further show that SAGE is complementary to training-time plug-ins. Overall, our results identify post-hoc sanitization of final unlearning updates as a practical and underexplored design axis for machine unlearning, and suggest that directly correcting the final deployed update can be an effective way to improve unlearning quality.

Bibliography

[1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[2] Karuna Bhaila, Minh-Hao Van, and Xintao Wu. Soft prompting for unlearning in large language models. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025.

[3] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy (IEEE S&P)*, 2021.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. Language models are few-shot learners. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Chengyi Cai, Zesheng Ye, Jiangchao Yao, Jianzhong Qi, Bo Han, Xiaolu Zhang, Feng Liu, and Jun Zhou. Per-parameter task arithmetic for unlearning in large language models. *arXiv preprint arXiv:2601.22030*, 2026.

[6] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association, 2021.

[7] Junhao Dong, Hao Zhu, Yifei Zhang, Xinghua Qu, Yew Soon Ong, and Piotr Koniusz. Machine unlearning via task simplex arithmetic. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[8] 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. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025.

[9] Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C. Lipton, J. Zico Kolter, and Pratyush Maini. Openunlearning: Accelerating llm unlearning via unified benchmarking of methods and metrics. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[10] Taha Entesari, Arman Hatami, Rinat Khaziev, Anil Ramakrishna, and Mahyar Fazlyab. Constrained entropic unlearning: A primal-dual framework for large language models. *arXiv preprint arXiv:2506.05314*, 2025.

[11] Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[12] Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and Tat seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations (ICLR)*, 2025.

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

[14] James Y. Huang, Wenxuan Zhou, Fei Wang, Fred Morstatter, Sheng Zhang, Hoifung Poon, and Muhao Chen. Offset unlearning for large language models. *Transactions on Machine Learning Research*, 2025.

[15] Yu-Ting Huang, Pei-Yuan Wu, and Chuan-Ju Wang. Eco: Efficient computational optimization for exact machine unlearning in deep neural networks. In *International Conference on Machine Learning Workshop (ICML Workshop)*, 2024.

[16] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

[17] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

[18] Hyo Seo Kim, Dongyoon Han, and Junsuk Choe. Negmerge: Sign-consensual weight merging for machine unlearning. In *International Conference on Machine Learning (ICML)*, 2025.

[19] 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, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning (ICML)*, 2024.

[20] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

[21] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. Rethinking machine unlearning for large language models. In *Nature Machine Intelligence (Nat. Mach. Intell.)*, 2024.

[22] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms. In *Conference on Language Modeling (COLM*, 2024.

[23] Anmol Mekala, Vineeth Dorna, Shreya Dubey, Abhishek Lalwani, David Koleczek, Mukund Rungta, Sadid Hasan, and Elita Lobo. Alternate preference optimization for unlearning factual knowledge in large language models. *arXiv preprint arXiv:2409.13474*, 2024.

[24] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[25] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations (ICLR)*, 2023.

[26] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *International Conference on Algorithmic Learning Theory (ALT)*, 2021.

[27] Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners. In *International Conference on Machine Learning (ICML)*, 2024.

[28] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[29] William F. Shen, Xinchi Qiu, Meghdad Kurmanji, Alex Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas D. Lane. Llm unlearning via neural activation redirection. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

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

[31] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

[33] Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q. Weinberger. Rethinking llm unlearning objectives: A gradient perspective and go beyond. In *International Conference on Learning Representations (ICLR)*, 2025.

[34] Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. Llm unlearning via loss adjustment with only forget data. In *International Conference on Learning Representations (ICLR)*, 2024.

[35] Yue Wang, Qizhou Wang, Feng Liu, Wei Huang, Yali Du, Xiaojiang Du, and Bo Han. Gru: Mitigating the trade-off between unlearning and retention for llms. In *International Conference on Machine Learning (ICML)*, 2025.

[36] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[37] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[38] Bo Yang. Ce-u: Cross entropy unlearning. *arXiv preprint arXiv:2503.01224*, 2025.

[39] Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. Exploring criteria of loss reweighting to enhance llm unlearning. In *International Conference on Machine Learning (ICML)*, 2025.

[40] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2): 100211, 2024.

[41] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[42] Dawen Zhang, Pamela Finckenberg-Broman, Thong Hoang, Shidong Pan, Zhenchang Xing, Mark Staples, and Xiwei Xu. Right to be forgotten in the era of large language models: Implications, challenges, and solutions. *AI and Ethics (AI Ethics)*, 2025.

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

[44] Duo Zhou, Yuji Zhang, Tianxin Wei, Ruizhong Qiu, Ke Yang, Xiao Lin, Cheng Qian, Jingrui He, Hanghang Tong, Chengxiang Zhai, Heng Ji, and Huan Zhang. Geometric-disentangelment unlearning. *arXiv preprint arXiv:2511.17100*, 2026.