FEATURE REFLECTION FOR MACHINE UNLEARNING WITHOUT RETAIN DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning aims to remove designated knowledge while preserving a model's remaining capabilities. This preservation becomes more challenging when retain data are unavailable and unlearning relies solely on the pretrained model and the forget data. To overcome this challenge, we revisit unlearning from a Bayesian perspective and identify two requirements: reducing fit to the forget data while preserving pretrained knowledge. Motivated by these requirements, we introduce *Feature Reflection* (FR), a simple mechanism that turns bounded-below loss minimization into theoretically grounded forgetting. Specifically, FR reflects the features of the model being unlearned with respect to their pretrained counterparts, perturbs the reflection with multiplicative Gaussian noise, and minimizes the loss on the noisy reflected features. Theoretical analysis establishes that reducing the loss on reflected features raises a lower bound on the forget loss, explicitly connecting bounded-below minimization to forgetting. Further analysis shows that noise perturbation induces a curvature-weighted regularizer on deviations from pretrained features, promoting knowledge preservation. With a simple design and theoretical support, FR outperforms competitive retain-free unlearning baselines over eight vision datasets under different data settings, as well as on the WMDP benchmark. Code will be released upon acceptance.

1 INTRODUCTION

Deep learning (LeCun et al., 2015; Goodfellow et al., 2016) has advanced through increasingly large models (Vaswani et al., 2017) and datasets (Deng et al., 2009; Schuhmann et al., 2022), producing powerful pretrained models widely reused across downstream applications (Dosovitskiy et al., 2021; Radford et al., 2021; Bommasani et al., 2021). This widespread reuse also creates a growing need to remove information from released models in response to privacy concerns (Bărbulescu & Triantafillou, 2024), copyright claims (Basaran et al., 2025), and deletion requests (Brophy & Lowd, 2021). Addressing this need, machine unlearning (Cao & Yang, 2015; Bourtoule et al., 2021) aims to erase the influence of designated forget data while preserving the model's remaining knowledge (Koloskova et al., 2025). Such preservation often relies on retain data from the original pretraining distribution, yet this distribution may be inaccessible for large pretrained models and difficult to approximate faithfully with a small substitute set (Cheng et al., 2026). This practical limitation leads to the retain-free setting, where unlearning relies only on the pretrained model and the forget set.

Under the retain-free setting, one natural strategy is to optimize directly on the forget set. Gradient ascent (GA) follows this principle by increasing the forget loss, thereby reducing the model's fit to the data to be removed (Golatkar et al., 2020; Tarun et al., 2023). This ascent, however, typically maximizes the cross-entropy loss, which is unbounded above and lacks a finite optimum. Without a finite optimum, the extent of forgetting is determined by the number of training steps rather than by the loss itself. As these steps accumulate, the model's remaining utility can degrade substantially under such unbounded ascent (Zhang et al., 2024; Fan et al., 2025; Mavrothalassitis et al., 2025).

To avoid unbounded ascent, Task Arithmetic (TA) (Ilharco et al., 2023) replaces loss maximization with post-hoc parameter negation. It first minimizes the forget loss to obtain a parameter change, termed a task vector, that improves fitting to the forget data. Negating this task vector then reverses the parameter change to induce forgetting. This negation, however, acts on parameters, whose relationship to predictions is nonlinear (Ortiz-Jimenez et al., 2023; Li et al., 2025; Kim et al., 2025;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Dong et al., 2025). Under this nonlinearity, further reductions in the forget loss during fine-tuning are not guaranteed to translate into further forgetting after negation. Consequently, TA secures a bounded-below minimization objective but lacks a guarantee linking its minimization to forgetting.

The unbounded maximization in GA and the lack of a guarantee linking minimization to forgetting in TA call for a new unlearning mechanism that provably connects bounded-below loss minimization to forgetting. To guide the design of this mechanism, we revisit unlearning from a Bayesian perspective, identifying two requirements for unlearning without retain data: reducing the model's fit to the forget data while preserving the knowledge encoded by the pretrained model.

Motivated by these two requirements, we propose *Feature Reflection* (FR), which moves reflection from a post-hoc operation on parameters to a feature transformation during training. Specifically, FR reflects the features produced by the model being unlearned with respect to the corresponding pretrained features, perturbs the reflected features with multiplicative Gaussian noise, and minimizes the classification loss on the resulting noisy reflected features. Despite the simplicity of FR, our theoretical analyses demonstrate that reducing the loss on the reflected features raises a lower bound on the forget loss, connecting bounded-below minimization to forgetting and addressing the first requirement. We further show that noise perturbation implicitly regularizes deviations from the pretrained features to promote knowledge preservation, addressing the second requirement.

With its simple design (two lines of core code in Figure 1) and theoretical grounding, FR outperforms competitive baselines on CLIP vision unlearning (Ilharco et al., 2022) under full-data, few-shot, and long-tailed settings and on the WMDP benchmark (Li et al., 2024). Further experimental analyses support the theoretical rationale of FR. Our contributions are summarized as follows:

i. We introduce feature reflection, a simple retain-free unlearning mechanism that connects bounded-below loss minimization to forgetting, with a theoretical guarantee that reducing the loss on reflected features raises a lower bound on the forget loss.
ii. We propose FR as this reflection mechanism augmented with multiplicative Gaussian noise perturbation and show that it implicitly induces a data-dependent, curvature-weighted regularizer on deviations from pretrained features, promoting knowledge preservation.
iii. We demonstrate that FR outperforms competitive retain-free methods on CLIP vision unlearning under full-data, few-shot, and long-tailed settings and on the WMDP benchmark, with further analyses supporting its forgetting and preservation mechanisms.

2 Related Work

Machine unlearning. Machine unlearning seeks to remove the influence of designated training data while preserving the model's remaining utility (Cao & Yang, 2015; Bourtoule et al., 2021). Such preservation typically relies on resources beyond the forget data. One resource is training-time preparation. SISA (Bourtoule et al., 2021) isolates training across data shards and retains checkpoints to reduce retraining cost, while certified removal (Guo et al., 2020) provides formal guarantees for linear classifiers trained with a perturbed loss. A second resource is retain data. Weight scrubbing (Golatkar et al., 2020) perturbs weights according to Fisher information of the retained data, SCRUB (Kurmanji et al., 2023) promotes disagreement with the original model on forget data and agreement on retain data, and SalUn (Fan et al., 2024) restricts updates to parameters selected by forget-gradient saliency while regularizing them with retain data. A third resource, stored training statistics, underlies Selective Synaptic Dampening (Foster et al., 2024), which modifies parameters according to Fisher importance estimated from forget and full training data. Our setting excludes all three resources, requiring preservation from only a pretrained model and its forget data.

Optimization-based unlearning. Optimizing on forget data provides a direct route to unlearning, and existing methods differ mainly in the objective they optimize. The most direct choice is GA (Golatkar et al., 2020; Tarun et al., 2023), which increases the forget loss to reduce fitting but leaves this loss unbounded above. Alternative objectives replace ascent with random labeling in Amnesiac ML (Graves et al., 2021), entropy maximization in EntUn (Jung, 2025), or input-sensitivity suppression in MU-Mis (Cheng et al., 2026). Rather than replacing ascent, language-model unlearning tempers it. Negative Preference Optimization (NPO) (Zhang et al., 2024) uses a bounded preference-based loss. SimNPO (Fan et al., 2025) removes its reference model, and Weighted Gradient Ascent (WGA) (Wang et al., 2025) reweights token-level updates by model confidence. All

these methods redesign the objective on forget data. FR instead keeps the standard classification loss and transforms the features on which it is minimized, yielding a lower bound on the forget loss and an implicit regularizer toward pretrained features.

Task-vector unlearning. TA (Ilharco et al., 2023) encodes the parameter change from fine-tuning as a task vector, whose negation suppresses the corresponding task. Subsequent methods improve the reliability of this negation along two lines. The first strengthens weight disentanglement through linearization in Linearized TA (Ortiz-Jimenez et al., 2023) or orthogonality regularization in OrthoReg (Liu et al., 2026). The second aggregates fine-tuning runs, merging sign-consistent task-vector components in NegMerge (Kim et al., 2025) or ensembling predictions over a task-vector simplex in Task Simplex (Dong et al., 2025). Linearization makes negation reflect the features exactly but confines the model to its tangent space at extra cost, while aggregation requires combining several fine-tuned models. FR instead reflects features before the linear head, achieving exact reflection in a single run without linearization and linking loss minimization directly to forgetting.

Representation-based unlearning. Representation-based methods suppress undesired knowledge or behavior by modifying internal features. Representation Misdirection for Unlearning (RMU) (Li et al., 2024) steers forget-data features toward a fixed random target, while Representation Rerouting (Zou et al., 2024) pushes harmful features toward orthogonality with their original counterparts. These targets are defined independently of the prediction loss, and both preserve capabilities by matching pretrained features on retain data. FR instead reflects each feature with respect to its pretrained counterpart and fits the reflected feature, which raises a lower bound on the forget loss. For preservation, FR replaces retain-data matching with noise-induced regularization.

3 METHOD

3.1 MOTIVATION

With large datasets and massive computation, pretrained models have acquired broad knowledge and capabilities, and are increasingly released for reuse across downstream applications. Adapting such a released model $\boldsymbol{\theta}_0$ to a specific application may require removing the knowledge associated with an undesired domain, carried by $\mathcal{D}_f$, while preserving knowledge learned from the remaining pretraining data $\mathcal{D}_r$. These retain data, however, are often not released alongside the model, and collecting a representative substitute is difficult given their scale and diversity. Without access to $\mathcal{D}_r$, separating knowledge to forget from knowledge to preserve is challenging. To make this separation explicit, we analyze a model trained only on $\mathcal{D}_r$ from a Bayesian perspective (Nguyen et al., 2020):

Proposition 1 *Let $\mathcal{D}_f$ and $\mathcal{D}_r$ denote disjoint subsets of the i.i.d. pretraining data $\mathcal{D}_f \cup \mathcal{D}_r$. Then*

$$\log p(\boldsymbol{\theta}|\mathcal{D}_r) = -\log p(\mathcal{D}_f|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}|\mathcal{D}_f, \mathcal{D}_r) + C, \tag{1}$$

where $C = \log p(\mathcal{D}_f, \mathcal{D}_r) - \log p(\mathcal{D}_r)$ is a constant independent of $\boldsymbol{\theta}$.

Prop. 1 (proof in § A.1) shows that maximizing the retain posterior $\log p(\boldsymbol{\theta}|\mathcal{D}_r)$ amounts to jointly lowering the model's fit to the forget data and maximizing the full posterior $\log p(\boldsymbol{\theta}|\mathcal{D}_f, \mathcal{D}_r)$. The first objective is directly optimizable using the available $\mathcal{D}_f$. The second, however, cannot be optimized without access to the full data. Fortunately, the pretrained parameters $\boldsymbol{\theta}_0$ are available and can be viewed as an approximate maximizer of the full posterior, making them a practical reference during unlearning. This reference, together with $\mathcal{D}_f$ motivates two requirements for unlearning without retain data: (R1) lowering the fit to $\mathcal{D}_f$ and (R2) preserving the knowledge carried by $\boldsymbol{\theta}_0$.

3.2 REFLECTING FEATURES FOR SATISFYING R1

Gradient Ascent (GA). A direct way to satisfy R1 is to maximize the loss on the forget data $\mathcal{D}_f$ by GA. GA on the cross-entropy loss, however, lacks the finite optimal value that anchors loss minimization. Without such an optimal value, the fitting of the model can be arbitrarily worsened. To make this precise, we consider the widely adopted setting of a linear classification head followed by the cross-entropy loss. Let $\boldsymbol{z} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ be the feature of an input $\boldsymbol{x}$ with label $y \in \{1, \dots, K\}$, and let $h(\boldsymbol{z}) = \boldsymbol{W}\boldsymbol{z}$ be the classification head with rows $\boldsymbol{w}_k$, where the bias is absorbed into $\boldsymbol{W}$

for simplicity and $\boldsymbol{W}$ is kept fixed[1]. The following proposition considers the corresponding cross-entropy loss $\ell(\boldsymbol{z}, y) = \log\left(1 + \sum_{k \neq y} e^{-(\boldsymbol{w}_y - \boldsymbol{w}_k)^T \boldsymbol{z}}\right)$.

Proposition 2 (GA is unbounded) *Suppose $\boldsymbol{w}_y - \boldsymbol{w}_k \neq \mathbf{0}$ for some $k \neq y$, then we have:*

$$\sup_{\boldsymbol{z}} \ell(\boldsymbol{z}, y) = +\infty. \tag{2}$$

Prop. 2 (proof in § A.2) indicates that the forget loss can grow arbitrarily large, leaving the extent of forgetting to be determined by the number of training steps rather than by the objective. Such unbounded growth may entail substantial model changes that disrupt pretrained knowledge, consistent with the collapse observed in prior work (Zhang et al., 2024; Mavrothalassitis et al., 2025).

Task Arithmetic (TA). GA satisfies R1 directly, but its unbounded objective lacks a finite optimal value at which forgetting can terminate. Gradient descent on the same loss, by contrast, is bounded below by zero. TA benefits from this asymmetry by running gradient descent to fit the forget data $\mathcal{D}_f$, then reflecting the learned parameters to undo that fit to induce unlearning. Specifically, starting from $\boldsymbol{\theta}_0$, fine-tuning with learning rate η updates the parameters as $\boldsymbol{\theta}_{t+1}^+ = \boldsymbol{\theta}_t^+ - \eta \boldsymbol{g}_t^+$, where $\boldsymbol{g}_t^+ = \nabla L_f(\boldsymbol{\theta}_t^+)$ is the forget-loss gradient. After T steps, the fine-tuned parameters are reflected about $\boldsymbol{\theta}_0$ to produce the unlearned model $\boldsymbol{\theta}_T^- = 2\boldsymbol{\theta}_0 - \boldsymbol{\theta}_T^+$. To see how this parameter reflection induces forgetting, we define the reflected iterate $\boldsymbol{\theta}_t^- = 2\boldsymbol{\theta}_0 - \boldsymbol{\theta}_t^+$ for each step t. Substituting the descent update into this definition gives:

$$\boldsymbol{\theta}_{t+1}^- = \boldsymbol{\theta}_t^- + \eta \boldsymbol{g}_t^+. \tag{3}$$

Parameter reflection thus converts each descent step on the fine-tuning trajectory into an ascent-like step on the reflected trajectory. This step resembles GA but uses a gradient computed at a different point. At $\boldsymbol{\theta}_t^-$, the steepest ascent direction is the local gradient $\boldsymbol{g}_t^- = \nabla L_f(\boldsymbol{\theta}_t^-)$, whereas TA transfers $\boldsymbol{g}_t^+$ from the fine-tuning iterate $\boldsymbol{\theta}_t^+$. As the two iterates move apart, these gradients can diverge, and the transferred step no longer guarantees an increase in the forget loss. The following proposition quantifies the forgetting achieved by this transferred gradient.

Proposition 3 (Unlearning effect of task negation) *Assume that L_f is β-smooth, and let $\gamma_t = \cos(\boldsymbol{g}_t^-, \boldsymbol{g}_t^+)$. Then, the per-step change in the forget loss along the reflected trajectory verifies*

$$L_f(\boldsymbol{\theta}_{t+1}^-) - L_f(\boldsymbol{\theta}_t^-) \geq \eta\gamma_t \|\boldsymbol{g}_t^-\| \|\boldsymbol{g}_t^+\| - \frac{\beta\eta^2}{2} \|\boldsymbol{g}_t^+\|^2. \tag{4}$$

Prop. 3 (proof in § A.3) reveals that, for small η, the forgetting gained at each step is dominated by $\eta\gamma_t \|\boldsymbol{g}_t^-\| \|\boldsymbol{g}_t^+\|$, whose sign follows the ascent alignment γ_t, the cosine similarity between the transferred gradient $\boldsymbol{g}_t^+$ and the local ascent direction $\boldsymbol{g}_t^-$. This alignment is exact at initialization, where both trajectories start from $\boldsymbol{\theta}_0$, giving $\gamma_0 = 1$ and making the first step an exact ascent step. As fine-tuning proceeds, however, $\boldsymbol{\theta}_t^+$ and $\boldsymbol{\theta}_t^-$ move apart in opposite direction from $\boldsymbol{\theta}_0$, and the nonlinearity of the network can make their gradients differ substantially, weakening this alignment. As γ_t approaches zero, the guaranteed increase in the forget loss vanishes. Once γ_t becomes negative, the lower bound itself turns negative, and the update may decrease the forget loss, potentially undoing part of the forgetting gained by earlier steps.

This analysis exposes both the strength and the limitation of TA. Its strength lies in replacing loss maximization with bounded loss minimization, which offers a more controlled route to forgetting. Its limitation lies in where the reflection is performed. TA reflects parameters, which the network maps to predictions nonlinearly. As fine-tuning proceeds, this nonlinearity can weaken the alignment γ_t, and the guaranteed forgetting weakens with it.

Reflecting features. To prevent this weakening without resorting to unbounded loss maximization, we seek a bounded objective whose minimization induces forgetting directly within the same model. Rather than reflecting the parameters after fine-tuning, we perform the reflection at the feature level during training. Let $\boldsymbol{z}_0 = f_{\boldsymbol{\theta}_0}(\boldsymbol{x})$ and $\boldsymbol{z} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ denote features from the pretrained model and the model being unlearned, respectively. We reflect $\boldsymbol{z}$ with respect to $\boldsymbol{z}_0$ to obtain:

$$\boldsymbol{z}_r = \boldsymbol{z}_0 + (\boldsymbol{z}_0 - \boldsymbol{z}) = 2\boldsymbol{z}_0 - \boldsymbol{z}, \tag{5}$$

[1] In CLIP the head is formed by the text encoder and fixed by construction. In a language model it is shared across all domains, so altering it would shift knowledge beyond the forget domain. We thus keep the head fixed.

and minimize the classification loss on the reflected feature $\boldsymbol{z}_r$. As $\boldsymbol{z}_r$ and $\boldsymbol{z}$ move by equal amounts in opposite directions, improving the prediction at $\boldsymbol{z}_r$ worsens the prediction on $\boldsymbol{z}$, increasing its forget loss. Crucially, both features come from the same model at each training step, and every update is computed and applied on the model being unlearned rather than through the cross-trajectory transfer required by TA. By avoiding this transfer, feature reflection admits a direct forgetting guarantee, established in the following proposition.

Proposition 4 (Minimizing the reflected loss induces forgetting) *Consider the cross-entropy loss* $\ell(\boldsymbol{z}, y) = \log(1 + \sum_{k \neq y} e^{-m_k(\boldsymbol{z})})$ *with margins* $m_k(\boldsymbol{z}) = (\boldsymbol{w}_y - \boldsymbol{w}_k)^\top \boldsymbol{z}$ *under a fixed linear head. Let* $M_0 = \max_{k \neq y} m_k(\boldsymbol{z}_0)$ *and reflected feature be* $\boldsymbol{z}_r = 2\boldsymbol{z}_0 - \boldsymbol{z}$*. Then*

$$\ell(\boldsymbol{z}, y) \geq \log\left(1 + \frac{(K-1)^2 e^{-2M_0}}{e^{\ell(\boldsymbol{z}_r, y)} - 1}\right), \quad \textit{and } \ell(\boldsymbol{z}_r, y) \to 0 \textit{ implies } \ell(\boldsymbol{z}, y) \to +\infty. \tag{6}$$

Prop.4 (proof in §A.4) shows that decreasing the reflected loss $\ell(\boldsymbol{z}_r, y)$ monotonically raises a lower bound on the forget loss $\ell(\boldsymbol{z}, y)$, and this bound grows large as $\ell(\boldsymbol{z}_r, y) \to 0$ ensuring that sufficient forgetting can be achieved. Unlike GA, which optimizes an unbounded objective, feature reflection achieves this forgetting by optimizing an objective bounded below by zero. TA also optimizes a bounded objective, yet it lacks this forgetting guarantee.

With the forgetting guarantee of feature reflection established, we examine why, under the same bounded minimization, feature reflection yields more reliable forgetting than TA. Each method forgets through its update, whose effectiveness depends on its alignment with the ascent direction of the forget loss. For a single sample, where L_f reduces to ℓ, this ascent alignment is $\gamma = \cos(\boldsymbol{g}^+, \boldsymbol{g}^-)$ for TA (Prop. 3), and $\gamma = \cos(\boldsymbol{g}, -\boldsymbol{g}_r)$ for reflecting features, with $\boldsymbol{g} = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{z}, y)$ and $\boldsymbol{g}_r = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{z}_r, y)$. Let $\boldsymbol{J}$ denote the feature Jacobian and $\boldsymbol{q}$, $\boldsymbol{q}_r$ the loss gradients with respect to $\boldsymbol{z}$, $\boldsymbol{z}_r$, with $\boldsymbol{J}^\pm$ and $\boldsymbol{q}^\pm$ their counterparts at $\boldsymbol{\theta}^\pm$. By the chain rule, both alignments take the form:

$$\text{TA: } \langle \boldsymbol{g}^-, \boldsymbol{g}^+ \rangle = (\boldsymbol{q}^-)^\top \boldsymbol{J}^- (\boldsymbol{J}^+)^\top \boldsymbol{q}^+ \quad \text{vs} \quad \text{Reflecting features: } \langle \boldsymbol{g}, -\boldsymbol{g}_r \rangle = \boldsymbol{q}^\top \boldsymbol{J} \boldsymbol{J}^\top \boldsymbol{q}_r, \tag{7}$$

where $-\boldsymbol{g}_r = \boldsymbol{J}^T \boldsymbol{q}_r$, as minimization and reflection each flip the sign. Both forms pair two different loss gradients, which is intrinsic to forgetting through descent on a bounded loss and thus does not distinguish the two methods. The distinction lies in the Jacobian product between these gradients, whose two factors we compare through the Jacobian alignment $\omega = \cos(\boldsymbol{J}^-, \boldsymbol{J}^+)$, the cosine similarity between the flattened Jacobians. TA computes its gradients at two different models, yielding the cross-model product $\boldsymbol{J}^-(\boldsymbol{J}^+)^\top$, which is not necessarily positive semidefinite and can reverse the alignment even when the loss gradients coincide, turning γ negative. Reflecting features computes both gradients through the same model, yielding the positive semidefinite product $\boldsymbol{J}\boldsymbol{J}^\top$ with $\omega = 1$ by construction, which keeps the alignment nonnegative whenever the two loss gradients coincide. The more reliable forgetting thus stems from the mechanism of reflecting features within a single model rather than parameters across two.

3.3 Noise Perturbation for Knowledge Preservation

Reflecting features satisfies the first requirement with a bounded objective that lowers the model's fit to $\mathcal{D}_f$. We now turn to the second requirement, preserving the knowledge carried by the pretrained model. Without access to retain data $\mathcal{D}_r$, predictions on it cannot be constrained directly. We therefore use the pretrained feature $\boldsymbol{z}_0$ as the reference for preservation and limit unnecessary deviation from it during forgetting. Rather than adding an explicit penalty, we impose this limit implicitly by perturbing the reflected displacement with element-wise multiplicative Gaussian noise,

$$\widetilde{\boldsymbol{z}}_r = \boldsymbol{z}_0 + (\boldsymbol{z}_0 - \boldsymbol{z}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{1}, \sigma^2 \boldsymbol{I}), \tag{8}$$

and minimize the loss over $\widetilde{\boldsymbol{z}}_r$. The following proposition shows that, in expectation, this reduces to the clean loss plus a curvature-weighted penalty on the deviation from the pretrained feature.

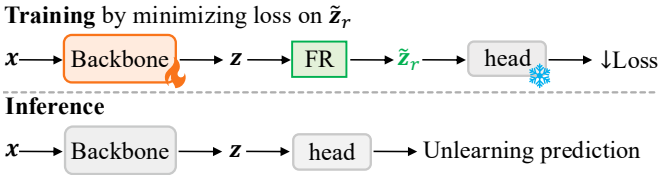
Proposition 5 (Implicit curvature-weighted feature regularization) *Let* $\Delta \boldsymbol{z} = \boldsymbol{z} - \boldsymbol{z}_0$ *and* $\boldsymbol{H} = \nabla^2_{\boldsymbol{z}_r} \ell(\boldsymbol{z}_r, y)$*, and denote by* H_{ii} *the* i*-th diagonal entry of* $\boldsymbol{H}$*. The expected noisy reflected loss admits the second-order expansion*

$$\mathbb{E}_{\boldsymbol{\epsilon}}[\ell(\widetilde{\boldsymbol{z}}_r, y)] = \ell(\boldsymbol{z}_r, y) + \frac{\sigma^2}{2} \sum_i H_{ii} (\Delta z_i)^2 + \mathcal{R}, \tag{9}$$

where $\mathcal{R}$ *collects the higher-order terms.*

Prop. 5 shows that, to second order, minimizing the loss on $\widetilde{\boldsymbol{z}}_r$ is equivalent to minimizing the loss on $\boldsymbol{z}_r$ with a regularization on the deviation $\Delta \boldsymbol{z}$ from the pretrained feature. The strength of this regularization on each Δz_i is adaptively controlled by the local curvature H_{ii} and the variance σ^2. Consequently, this data-dependent regularization discourages unnecessary deviation from the pretrained representation, helping preserve the knowledge encoded by the pretrained model.

3.4 Overall Method



Pytorch-style pseudo code for FR

```
import torch
def feature_reflection(z, z_0, σ):
  ϵ = torch.randn_like(z) * σ + 1
  z̃_r = z_0 + (z_0 - z) * ϵ
  return z̃_r
```

Figure 1: Left: training minimizes loss on noisy reflected features; inference uses the unlearned features directly. Right: the feature transformation requires only two lines of PyTorch-style code.

The preceding analyses establish how reflecting the features induces forgetting and how the noisy extension preserves pretrained knowledge. Building on these results, the overall method reduces to a single noisy feature transformation (Figure 1), which we refer to as Feature Reflection (FR). FR operates on two features of each sample $(\boldsymbol{x}, y) \in \mathcal{D}_f$, the feature to be unlearned $\boldsymbol{z} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ and the pretrained feature $\boldsymbol{z}_0 = f_{\boldsymbol{\theta}_0}(\boldsymbol{x})$. The pretrained feature does not change during unlearning, and it can be extracted once and stored in advance, avoiding repeated forward passes through $f_{\boldsymbol{\theta}_0}$. Using $\boldsymbol{z}$ and $\boldsymbol{z}_0$, we construct $\widetilde{\boldsymbol{z}}_r$ as in equation 8 and minimize the classification loss $\ell(\widetilde{\boldsymbol{z}}_r, y)$. Within this construction, the reflection addresses R1 by raising a lower bound on the forget loss (Prop. 4), and the noise perturbation addresses R2 by imposing a curvature-weighted penalty on deviation from the pretrained feature (Prop. 5). After unlearning, the feature $\boldsymbol{z}$ is used directly for inference without reflection or noise, leaving the inference cost unchanged.

4 Experiments

We evaluate FR on vision unlearning with CLIP models over eight datasets and WMDP with Zephyr-7B-β, measuring forgetting and preservation without using retain data for parameter updates.

4.1 Experimental Setup

Datasets and models. For vision, we adopt the task unlearning setting of prior work (Ilharco et al., 2023; Ortiz-Jimenez et al., 2023; Dong et al., 2025), using CLIP (Radford et al., 2021). Each of eight datasets serves as a separate forget task: Stanford Cars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Stallkamp et al., 2011), MNIST (Lecun et al., 1998), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2010), and SVHN (Netzer et al., 2011). ImageNet (Deng et al., 2009) measures retained generalization. We consider full-data, few-shot (5, 10, or 20 examples per class), and long-tailed settings (imbalance ratios of 0.01, 0.05, or 0.1). For language, we use Zephyr-7B-β (Tunstall et al., 2024), evaluating forgetting on WMDP (Li et al., 2024) and retention on MMLU (Hendrycks et al., 2021).

Baselines. Vision baselines include Random Labeling (RL), Entropy Maximization (EM), Gradient Ascent (GA), SalUn (Fan et al., 2024), TA (Ilharco et al., 2023), OrthoReg (OR) (Liu et al., 2026), Linearized TA (LTA) (Ortiz-Jimenez et al., 2023), NegMerge (NM) (Kim et al., 2025), and Task Simplex (TS) (Dong et al., 2025). Language baselines include GA, TA, NPO (Zhang et al., 2024), SimNPO (Fan et al., 2025), WGA (Wang et al., 2025), and RMU (Li et al., 2024). All methods use only forget data for training; retain-data losses are omitted where present in original formulations.

Evaluation and model selection. Following prior protocols (Ilharco et al., 2023; Ortiz-Jimenez et al., 2023; Dong et al., 2025), retain data are used only for validation and evaluation. For CLIP, we scale each method's learned parameter change by $\lambda \in \{0.05, 0.10, \ldots, 1.50\}$ after training. We select the λ minimizing forget validation accuracy while preserving at least 95% of pretrained accuracy on a small ImageNet training subset; the official ImageNet validation split is reserved for

testing. For WMDP, we use 20% of questions for validation and 80% for testing, with MMLU's validation and test splits measuring retention. Hyperparameters are selected by the validation harmonic mean of chance-normalized forgetting and retention scores. Results are averaged over five seeds.

Implementation. Following Ilharco et al. (2023), we update the CLIP image encoder while keeping the text encoder and its zero-shot head fixed. Training uses AdamW (Loshchilov & Hutter, 2019) with batch size 128, peak learning rate 1e-5, weight decay 0.1, and cosine scheduling. Language experiments build on OpenUnlearning (Dorna et al., 2025), using batch size 32, learning rate 1e-5, weight decay 1e-3, and linear scheduling. We fix $\sigma = 2$ across all vision datasets and data settings, and $\sigma = 0.01$ for language experiments. Additional experimental and dataset details are in § B.

4.2 Main Results

Table 1: Full-data unlearning of CLIP models: average forget accuracy over 8 datasets and retain accuracy on ImageNet (%). Model selection preserves at least 95% of pretrained retain accuracy on validation data. Results are averaged over five trials. †Requires 30 fine-tuned models.

Method	ViT-Base/32		ViT-Base/16		ViT-Large/14	
	Forget (↓)	Retain (↑)	Forget (↓)	Retain (↑)	Forget (↓)	Retain (↑)
Pre-trained Model	48.09	63.33	55.46	68.33	65.22	75.53
Random Label (RL)	32.22	60.57	28.22	65.23	22.31	72.25
Entropy Maximization (EM)	26.27	60.47	23.93	65.21	20.02	72.81
Gradient Ascent (GA)	12.28	60.61	12.73	65.13	12.82	72.88
SalUn	12.19	60.57	12.30	**65.57**	12.85	73.29
Task Arithmetic (TA)	22.38	60.37	21.50	65.31	18.26	72.27
OrthoReg (OR)	16.60	60.36	17.10	65.38	12.83	72.42
NegMerge† (NM)	16.88	59.86	18.92	65.05	16.11	71.92
Task Simplex† (TS)	14.96	60.21	12.74	65.41	10.06	72.61
Linearized TA (LTA)	10.17	60.16	9.35	65.11	8.28	71.88
FR	**6.30**	**60.66**	**4.55**	65.46	**4.00**	**73.34**

Full-data unlearning. Table 1 compares FR with competing methods across three CLIP architectures: ViT-Base/32, ViT-Base/16, and ViT-Large/14. Results report average forget and retain accuracies over eight separate dataset-unlearning tasks. All methods preserve approximately 95% of pretrained retain accuracy, while FR achieves the lowest forget accuracy. FR outperforms NegMerge (NM) and Task Simplex (TS) despite both requiring 30 fine-tuned models. It also outperforms Linearized TA (LTA), which incurs higher computational and GPU memory costs during training (Figure 2).

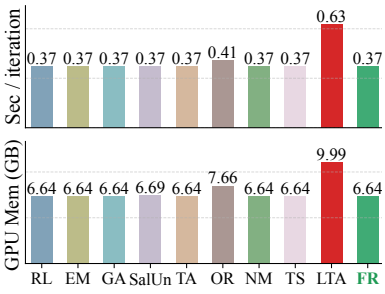


Figure 2: Training time (s/iteration) and peak GPU memory for ViT-Base/32 on V100 (32GB), averaged over eight datasets.

Few-shot unlearning. Building on the full-data results, we evaluate the more practical few-shot setting, where collecting forget data is difficult and only limited examples per class are available. Table 2 (left) compares FR with baselines in the 5-, 10-, and 20-shot settings. FR achieves the lowest average forget accuracy, while all methods preserve approximately 95% of pretrained retain accuracy. With only 10 training examples per class, FR achieves 9.93% average forget accuracy, outperforming LTA's 10.17% in the full-data setting with approximately 280 examples per class on average (Table 1).

Long-tailed unlearning. We further evaluate a practical scenario where forget data follow a long-tailed class distribution. Table 2 (right) compares FR with baselines at imbalance ratios $\rho \in \{0.01, 0.05, 0.1\}$. FR achieves the lowest average forget accuracy across all ratios, demonstrating its effectiveness under class imbalance.

LLM Unlearning on WMDP. To evaluate FR beyond vision, we conduct experiments on WMDP-Cyber and WMDP-Bio using Zephyr-7B-β. We report forget accuracy, MMLU accuracy for retention, and the harmonic mean of forgetting and retention scores in Table 3. FR achieves the highest

Table 2: Few-shot and long-tailed unlearning of CLIP (ViT-Base/32) across shot counts and imbalance ratios, respectively: average forget accuracy over 8 datasets and retain accuracy on ImageNet (%). Model selection preserves at least 95% of pretrained retain accuracy on validation data. Results are averaged over five trials. (F: Forget, R: Retain)

Method	Few-shot unlearning						Long-tailed unlearning					
	5 shots		10 shots		20 shots		$\rho = 0.01$		$\rho = 0.05$		$\rho = 0.1$	
	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)
Pre-trained Model	48.09	63.33	48.09	63.33	48.09	63.33	48.09	63.33	48.09	63.33	48.09	63.33
Random Label	26.67	60.18	26.50	60.27	29.00	60.31	34.20	60.50	33.86	60.40	33.32	60.45
Entropy Maximization	26.47	60.57	26.68	60.28	26.70	60.63	25.53	60.37	26.65	60.55	26.22	60.52
Gradient Ascent	13.57	60.51	12.83	60.33	12.51	60.55	13.91	60.03	13.40	60.46	13.58	60.46
SalUn	13.77	60.47	13.75	60.41	12.23	60.52	14.31	60.11	13.42	60.55	13.51	60.42
Task Arithmetic	23.73	60.41	23.97	**60.75**	23.88	60.43	24.37	60.52	22.80	60.28	23.08	60.35
OrthoReg	20.76	61.11	20.83	60.00	18.60	60.66	17.40	60.27	16.72	60.43	16.76	60.47
NegMerge	20.66	59.77	20.31	59.76	19.30	59.88	19.11	59.91	18.33	59.86	18.08	60.02
Task Simplex	16.75	60.07	15.70	59.93	15.28	60.15	16.61	60.11	15.82	60.12	15.33	60.01
Linearized TA	15.48	60.08	14.26	59.82	12.48	59.77	13.96	60.03	12.05	60.11	12.13	60.08
FR	**12.71**	**60.62**	**9.93**	60.54	**8.07**	**60.76**	**11.36**	**60.63**	**9.38**	**60.69**	**8.36**	**60.78**

Table 3: Unlearning on WMDP based on Zephyr-7B-β. Results are averaged over five trials.

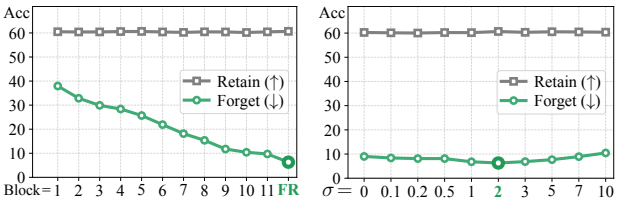
Method	Forget Cyber ↓	MMLU ↑	Harmonic Mean ↑	Forget Bio ↓	MMLU ↑	Harmonic Mean ↑
Original	44.27	58.92	0	66.63	58.92	0
Gradient Ascent	34.57	52.54	61.94	30.83	45.21	69.76
Task Arithmetic	30.59	48.27	69.04	40.12	45.71	61.43
NPO	**26.40**	40.38	59.86	27.77	36.94	49.96
SimNPO	35.60	50.14	56.00	36.41	45.55	65.40
WGA	31.02	49.03	68.33	**25.94**	34.56	43.47
RMU	29.80	49.41	73.15	34.90	41.84	59.92
FR	28.84	**53.16**	**81.45**	30.46	**46.14**	**72.04**

MMLU accuracy and harmonic mean, demonstrating the best balance between forgetting and retention among the evaluated methods and supporting its applicability to LLMs.

4.3 Ablation Studies

Where to apply FR. We evaluate FR at different network depths in ViT-Base/32, averaging forget and retain accuracies over eight datasets. Figure 3a shows that applying reflection deeper in the network generally lowers forget accuracy. Reflection immediately before the classification layer achieves the lowest forget accuracy, supporting our default placement.

Effect of σ. We vary σ from 0 to 10, with $\sigma = 0$ corresponding to FR without noise. Figure 3b shows that $\sigma = 2$ yields the lowest forget accuracy, supporting its use across all vision settings.



(a) Reflection depth. (b) Noise standard deviation σ.

Figure 3: Forget and retain accuracies across reflection depths and noise standard deviations for full-data unlearning with ViT-Base/32, averaged over eight datasets.

Different variants. Table 4 highlights the contributions of reflection and noise perturbation. Even without noise, FR achieves 9.01% forget accuracy, compared with 22.38% for TA. Adding noise further lowers forget accuracy to 6.30%. Full FR achieves the best forgetting and retention across all tested variants, including alternative noise distributions, a feature deviation penalty $\|\boldsymbol{z} - \boldsymbol{z}_0\|_2^2$ (FeatReg), and GA and TA combined with FeatReg or a parameter deviation penalty $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2$ (ParamReg).

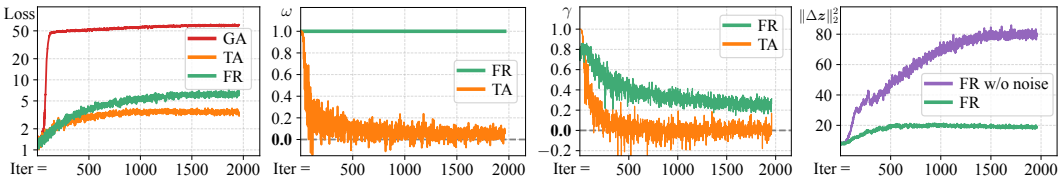
4.4 ANALYSES

Understanding forgetting in FR. Figure 4a compares the unlearning loss of the models used for inference. GA rapidly drives up the loss, while the increase of TA soon plateaus. FR sustains an increase through bounded-below minimization. Figures 4b and 4c reveal a key distinction between TA and FR. TA computes fitting gradients at $\boldsymbol{\theta}^{+}$ but induces forgetting at $\boldsymbol{\theta}^{-}$. The Jacobian alignment ω between these two models decays toward zero (Figure 4b) and the ascent alignment γ falls to around zero with negative excursions (Figure 4c), indicating that the induced forgetting direction becomes nearly orthogonal to, or opposes, forget-loss ascent. Further fitting can therefore become ineffective or locally counterproductive for forgetting. FR avoids this mismatch, as the feature being unlearned and its reflected counterpart share the same encoder Jacobian, fixing ω at one. Its forgetting direction remains positively aligned with forget-loss ascent, complementing our theoretical guarantee that reducing the loss on reflected features raises a lower bound on the forget loss.

Table 4: Forget and retain acc. across variants for full data ViT-Base/32 unlearning, averaged over eight datasets.

Method	Forget ↓	Retain ↑
GA	12.28	60.61
GA + FeatReg	10.84	60.37
GA + ParamReg	12.12	60.48
TA	22.38	60.37
TA + FeatReg	18.58	60.13
TA + ParamReg	21.87	60.28
FR w/o noise	9.01	60.25
FR w/ uniform noise	7.75	60.12
FR w/ dropout	7.13	60.31
FR w/o noise + FeatReg	8.52	60.36
FR	**6.30**	**60.66**

Understanding preservation in FR. Figure 4d shows that feature deviation grows without noise but remains much smaller with multiplicative Gaussian noise. This supports Prop. 5 that noise implicitly regularizes deviations from pretrained features, promoting knowledge preservation.



(a) Unlearning loss. (b) Jacobian alignment ω. (c) Ascent alignment γ. (d) Feature deviation.

Figure 4: Training dynamics on SUN397 with ViT-Base/32: (a) unlearning loss, (b) Jacobian alignment, (c) ascent alignment, and (d) feature deviation. See § C for Stanford Cars results.

Forgetting–retention trade-off. Figure 5 compares FR and TA as λ scales their learned parameter changes after training. Increasing λ strengthens forgetting but reduces retention for both methods. However, across most of the evaluated range, FR achieves lower forget accuracy and higher retain accuracy than TA, demonstrating a better forgetting–retention trade-off.

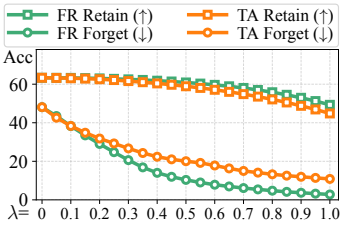


Figure 5: Forget and retain acc. over parameter scales λ on ViT-Base/32, averaged over eight datasets in the full-data setting.

5 CONCLUSION

We introduced Feature Reflection (FR), a simple unlearning mechanism that connects bounded-below loss minimization to forgetting without retain data during training. By reflecting features with respect to their pretrained counterparts, FR establishes a direct theoretical connection: reducing the loss on reflected features raises a lower bound on the forget loss. Multiplicative Gaussian noise complements this mechanism by inducing, to second order, a data-dependent curvature-weighted regularizer on feature deviations, promoting knowledge preservation. Experiments with CLIP models under full-data, few-shot, and long-tailed settings and on WMDP demonstrate a better forgetting–retention trade-off than competitive baselines. These results establish FR as a simple, theoretically grounded approach to balancing forgetting and preservation for unlearning without retain data.

AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing, including improving grammar, clarity and readability. We have not used generative AI tools for retrieval or discovery, research ideation or execution, drafting sections of the paper and proving mathematical claims. Generating synthetic datasets is not applicable to this work. We have reviewed all AI-assisted text to ensure its accuracy and consistency with our intended meaning and findings. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the method pipeline and PyTorch-style pseudocode in Figure 1. The experimental settings, datasets, and implementation details are described in Sec. 4 and §B. All proofs are provided in §A. We report results averaged over five random seeds.

REFERENCES

Umit Yigit Basaran, Sk Miraj Ahmed, Amit Roy-Chowdhury, and Basak Guler. A certified unlearning approach without access to source data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=8lt5776GLB.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pp. 141–159. IEEE, 2021.

Jonathan Brophy and Daniel Lowd. Machine unlearning for random forests. In *International conference on machine learning*, pp. 1092–1104. PMLR, 2021.

George-Octavian Bărbulescu and Peter Triantafillou. To each (Textual sequence) its own: Improving memorized-data unlearning in large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3003–3023. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/barbulescu24a.html.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and trends in Machine Learning*, 8(3-4):231–357, 2015.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Xinwen Cheng, Zhehao Huang, Wenxing Zhou, Zhengbao He, Ruikai Yang, Yingwen Wu, and Xiaolin Huang. Remaining-data-free machine unlearning by suppressing sample contribution. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3iw5t2W41F.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. *Advances in neural information processing systems*, 33:18613–18624, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Junhao Dong, Hao Zhu, Yifei Zhang, Xinghua Qu, Yew-Soon Ong, and Piotr Koniusz. Machine unlearning via task simplex arithmetic. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=pe92NSPLE8.

Vineeth Dorna, Anmol Reddy Mekala, Wenlong Zhao, Andrew McCallum, J Zico Kolter, Zachary Chase Lipton, and Pratyush Maini. Openunlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=Gy67Zh5X1i.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*, volume 2024, pp. 53643–53673, 2024.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=JbvSQm5h1l.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12043–12051, 2024.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9301–9309. IEEE, 2020.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11516–11524, 2021.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/guo20c.html.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=CZZFRxbOLC.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Dahuin Jung. Entun: Mitigating the forget-retain dilemma in unlearning via entropy. *ICT Express*, 2025.

Hyo Seo Kim, Dongyoon Han, and Junsuk Choe. Negmerge: Sign-consensual weight merging for machine unlearning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ZbWXovStjD.

Anastasia Koloskova, Youssef Allouah, Animesh Jha, Rachid Guerraoui, and Sanmi Koyejo. Certified unlearning for neural networks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=3rWQlV3s1I.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987, 2023.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Hongkang Li, Yihua Zhang, Shuai Zhang, Pin-Yu Chen, Sijia Liu, and Meng Wang. When is task vector provably effective for model editing? a generalization analysis of nonlinear transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vRvVVb0NAz.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam Alfred Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=xlr6AUDuJz.

Shangge Liu, Yuehan Yin, Lei Wang, Qi Fan, Yinghuan Shi, Wenbin Li, Yang Gao, and Dacheng Tao. Understanding and enforcing weight disentanglement in task arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28744–28753, June 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Ioannis Mavrothalassitis, Pol Puigdemont, Noam Itzhak Levi, and Volkan Cevher. Ascent fails to forget. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KBJSV1XApq.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning. *Advances in Neural Information Processing Systems*, 33:16025–16036, 2020.

Yao Ni and Piotr Koniusz. Nice: Noise-modulated consistency regularization for data-efficient gans. *Advances in Neural Information Processing Systems*, 36:13773–13801, 2023.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=0A9f2jZDGW.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pp. 1453–1460. IEEE, 2011.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE transactions on neural networks and learning systems*, 35(9):13046–13055, 2023.

Lewis Tunstall, Edward Emanuel 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 *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aKkAwZB6JV.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=huo8MqVH6t.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=MXLBXjQkmb.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=IbIB8SBKFV.

A PROOFS

A.1 PROOF FOR PROPOSITION 1

From Bayes' rule:

$$p(\boldsymbol{\theta}|\mathcal{D}_f, \mathcal{D}_r) = \frac{p(\mathcal{D}_f, \mathcal{D}_r|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D}_f, \mathcal{D}_r)}. \tag{10}$$

Since data is i.i.d. $\mathcal{D}_f$ and $\mathcal{D}_r$ are conditionally independent given $\boldsymbol{\theta}$, so

$$p(\mathcal{D}_f, \mathcal{D}_r|\boldsymbol{\theta}) = p(\mathcal{D}_f|\boldsymbol{\theta})p(\mathcal{D}_r|\boldsymbol{\theta}). \tag{11}$$

Therefore,

$$p(\boldsymbol{\theta}|\mathcal{D}_f, \mathcal{D}_r) = \frac{p(\mathcal{D}_f|\boldsymbol{\theta})p(\mathcal{D}_r|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D}_f, \mathcal{D}_r)} = \frac{p(\mathcal{D}_f|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D}_r)p(\mathcal{D}_r)}{p(\mathcal{D}_f, \mathcal{D}_r)} \tag{12}$$

Taking logarithms and rearranging,

$$\log p(\boldsymbol{\theta}|\mathcal{D}_r) = -\log p(\mathcal{D}_f|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}|\mathcal{D}_f, \mathcal{D}_r) + C \tag{13}$$

where

$$C = \log p(\mathcal{D}_f, \mathcal{D}_r) - \log p(\mathcal{D}_r), \tag{14}$$

which is independent of $\boldsymbol{\theta}$.

A.2 PROOF OF PROPOSITION 2

We define the margins $m_k(\boldsymbol{z}) = (\boldsymbol{w}_y - \boldsymbol{w}_k)^\top \boldsymbol{z}$ of the true class over each class k, so that $m_y(\boldsymbol{z}) = 0$, and

$$\ell(\boldsymbol{z}, y) \;=\; \log\Big(\sum_{k=1}^{K} e^{-m_k(\boldsymbol{z})}\Big) \;=\; \log\Big(1 + \sum_{k\neq y} e^{-m_k(\boldsymbol{z})}\Big). \tag{15}$$

Fix any $c \neq y$ with $\boldsymbol{w}_y - \boldsymbol{w}_c \neq \mathbf{0}$. Retaining only the term $k = c$ in the sum,

$$\ell(\boldsymbol{z}, y) \;\geq\; \log e^{-m_c(\boldsymbol{z})} \;=\; -(\boldsymbol{w}_y - \boldsymbol{w}_c)^\top \boldsymbol{z}. \tag{16}$$

The right hand side is a nonzero linear functional of $\boldsymbol{z}$ and is therefore unbounded above on $\mathbb{R}^d$, whence $\sup_{\boldsymbol{z}} \ell(\boldsymbol{z}, y) = +\infty$ and the maximization admits no solution.

A.3 PROOF FOR PROPOSITION 3

Since L_f is β-smooth, the descent lemma (Nesterov, 2013; Bubeck, 2015) gives, for any $\boldsymbol{\theta}$ and update $\Delta\boldsymbol{\theta}$,

$$L_f(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}) \geq L_f(\boldsymbol{\theta}) + \langle \nabla L_f(\boldsymbol{\theta}), \Delta\boldsymbol{\theta} \rangle - \frac{\beta}{2}\|\Delta\boldsymbol{\theta}\|^2. \tag{17}$$

For the reflected TA trajectory, $\boldsymbol{\theta}^-_{t+1} = \boldsymbol{\theta}^-_t + \eta \boldsymbol{g}^+_t$ where $\boldsymbol{g}^-_t = \nabla L_f(\boldsymbol{\theta}^-_t)$. Applying the smoothness inequality at $\boldsymbol{\theta}^-_t$ with $\Delta\boldsymbol{\theta} = \eta\boldsymbol{g}^+_t$ gives:

$$L_f(\boldsymbol{\theta}^-_{t+1}) - L_f(\boldsymbol{\theta}^-_t) \geq \eta\langle \boldsymbol{g}^-_t, \boldsymbol{g}^+_t \rangle - \frac{\beta\eta^2}{2}\|\boldsymbol{g}^+_t\|^2. \tag{18}$$

By the definition of the cosine similarity $\gamma_t = \frac{\langle \boldsymbol{g}^-_t, \boldsymbol{g}^+_t \rangle}{\|\boldsymbol{g}^-_t\|\|\boldsymbol{g}^+_t\|}$, we have $\langle \boldsymbol{g}^-_t, \boldsymbol{g}^+_t \rangle = \gamma_t \|\boldsymbol{g}^-_t\|\|\boldsymbol{g}^+_t\|$. Substituting this identity yields

$$L_f(\boldsymbol{\theta}^-_{t+1}) - L_f(\boldsymbol{\theta}^-_t) \geq \eta\gamma_t\|\boldsymbol{g}^-_t\|\|\boldsymbol{g}^+_t\| - \frac{\beta\eta^2}{2}\|\boldsymbol{g}^+_t\|^2. \tag{19}$$

A.4 Proof for Proposition 4

Recall that $\ell(\boldsymbol{z}, y) = \log\left(1+\sum_{k\neq y} e^{-m_k(\boldsymbol{z})}\right)$, with $m_k(\boldsymbol{z}) = (\boldsymbol{w}_y - \boldsymbol{w}_k)^\top \boldsymbol{z}$. Since $\boldsymbol{z}_r = 2\boldsymbol{z}_0 - \boldsymbol{z}$, linearity of the classifier gives

$$m_k(\boldsymbol{z}_r) = 2m_k(\boldsymbol{z}_0) - m_k(\boldsymbol{z}). \tag{20}$$

Hence,

$$e^{\ell(\boldsymbol{z},y)} - 1 = \sum_{k\neq y} e^{-m_k(\boldsymbol{z})} = \sum_{k\neq y} e^{-2m_k(\boldsymbol{z}_0)} e^{m_k(\boldsymbol{z}_r)}. \tag{21}$$

Let $M_0 = \max_{k\neq y} m_k(\boldsymbol{z}_0)$. Since $m_k(\boldsymbol{z}_0) \leq M_0$, for every $k \neq y$,

$$e^{\ell(\boldsymbol{z},y)} - 1 \geq e^{-2M_0} \sum_{k\neq y} e^{m_k(\boldsymbol{z}_r)}. \tag{22}$$

Meanwhile,

$$e^{\ell(\boldsymbol{z}_r,y)} - 1 = \sum_{k\neq y} e^{-m_k(\boldsymbol{z}_r)}. \tag{23}$$

Applying the Cauchy-Schwarz inequality to the $K-1$ positive terms gives:

$$\Big(\sum_{k\neq y} e^{m_k(\boldsymbol{z}_r)}\Big)\Big(\sum_{k\neq y} e^{-m_k(\boldsymbol{z}_r)}\Big) \geq (K-1)^2. \tag{24}$$

Substituting Eq.23 into Eq. 24, we have:

$$\sum_{k\neq y} e^{m_k(\boldsymbol{z}_r)} \geq \frac{(K-1)^2}{e^{\ell(\boldsymbol{z}_r,y)} - 1}. \tag{25}$$

Combining the inequalities in Eq. 22 and 25 yields

$$e^{\ell(\boldsymbol{z},y)} - 1 \geq \frac{(K-1)^2 e^{-2M_0}}{e^{\ell(\boldsymbol{z}_r,y)} - 1}. \tag{26}$$

Adding 1 and taking $\log$ on both sides gives:

$$\ell(\boldsymbol{z}, y) \geq \log\left(1 + \frac{(K-1)^2 e^{-2M_0}}{e^{\ell(\boldsymbol{z}_r,y)} - 1}\right). \tag{27}$$

The right-hand side is monotonically decreasing in $\ell(\boldsymbol{z}_r, y)$. Hence, decreasing the reflected loss raises the lower bound on $\ell(\boldsymbol{z}, y)$. Moreover, as $\ell(\boldsymbol{z}_r, y) \to 0$, $e^{\ell(\boldsymbol{z}_r,y)} - 1 \to 0$, and therefore the lower bound diverges to $+\infty$.

A.5 Proof for Proposition 5

The proof is motivated by Ni & Koniusz (2023). Let $\boldsymbol{\epsilon} = \mathbf{1} + \boldsymbol{\xi}$, with $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$. Since $\boldsymbol{z}_r = \boldsymbol{z}_0 - \Delta\boldsymbol{z}$,

$$\widetilde{\boldsymbol{z}}_r = \boldsymbol{z}_r - \Delta\boldsymbol{z} \odot \boldsymbol{\xi}. \tag{28}$$

Let $\delta\boldsymbol{z} = -\Delta\boldsymbol{z} \odot \boldsymbol{\xi}$. Expanding the loss around $\boldsymbol{z}_r$ gives

$$\ell(\boldsymbol{z}_r + \delta\boldsymbol{z}, y) = \ell(\boldsymbol{z}_r, y) + \nabla_{\boldsymbol{z}_r}\ell(\boldsymbol{z}_r, y)^\top \delta\boldsymbol{z} + \frac{1}{2}\delta\boldsymbol{z}^\top \boldsymbol{H}\delta\boldsymbol{z} + R(\delta\boldsymbol{z}). \tag{29}$$

Because $\mathbb{E}[\boldsymbol{\xi}] = \mathbf{0}$, we have $\mathbb{E}[\delta\boldsymbol{z}] = \mathbf{0}$, and thus the first-order term vanishes in expectation. For the quadratic term,

$$\mathbb{E}[\delta\boldsymbol{z}^\top \boldsymbol{H}\delta\boldsymbol{z}] = \sum_{i,j} \boldsymbol{H}_{ij}\Delta z_i \Delta z_j \mathbb{E}[\xi_i\xi_j] = \sigma^2 \sum_i \boldsymbol{H}_{ii}(\Delta z_i)^2, \tag{30}$$

where the second equality follows from the independence of the Gaussian noise, for which $\mathbb{E}[\xi_i\xi_j] = 0$ for $i \neq j$ and $\mathbb{E}[\xi_i^2] = \sigma^2$. Taking expectation therefore yields

$$\mathbb{E}_{\boldsymbol{\epsilon}}[\ell(\widetilde{\boldsymbol{z}}_r, y)] = \ell(\boldsymbol{z}_r, y) + \frac{\sigma^2}{2}\sum_i \boldsymbol{H}_{ii}(\Delta z_i)^2 + \mathcal{R}, \tag{31}$$

where $\mathcal{R} = \mathbb{E}[R(\delta\boldsymbol{z})]$ contains the higher-order contribution.

B Additional Experimental Details

We detail the data splits, model-selection protocols, baselines, and ablation variants. All methods use only forget data to compute training losses and parameter updates; retain-data losses are omitted where present in the original formulations. Forget and retain validation data guide model selection, while test data are reserved for final evaluation. Results are averaged over five random seeds.

B.1 Vision Unlearning

B.1.1 Datasets and splits.

Full-data unlearning. Following prior task-arithmetic studies (Ilharco et al., 2023; Ortiz-Jimenez et al., 2023; Dong et al., 2025), we evaluate CLIP on eight separate forget tasks, unlearning one dataset per experiment. We adopt the dataset loaders and validation-split procedure of Task Arithmetic (Ilharco et al., 2023). Table 5 lists the resulting splits and training epochs, which yield approximately 2,000 updates per full-data run. For retention, we use 5,000 randomly sampled ImageNet training images for validation and the official 50,000-image validation split for testing.

Table 5: Number of images per split and training epochs for each forget dataset in the full-data vision setting.

Dataset	Train	Validation	Test	Epochs
Stanford Cars	7,330	814	8,041	35
DTD	3,384	376	1,880	76
EuroSAT	21,600	2,700	2,700	12
GTSRB	23,976	2,664	12,630	11
MNIST	55,000	5,000	10,000	5
RESISC45	17,010	1,890	6,300	15
SUN397	17,865	1,985	19,850	14
SVHN	68,257	5,000	26,032	4

Few-shot unlearning. We randomly select $s \in \{5, 10, 20\}$ training examples per class and repeatedly sample from this fixed subset so that each epoch contains the same number of training examples as in the full-data setting. We retain the same number of epochs and the same validation and test splits.

Long-tailed unlearning. We rank classes by decreasing training-set size and sample n_r images without replacement from the class at rank $r \in \{0, \dots, K-1\}$, where

$$n_r = \min\Big\{N_r, \max\big[1, \mathrm{round}\big(N_{\max}\rho^{r/(K-1)}\big)\big]\Big\}.$$

Here, N_r is the available class count, $N_{\max} = \max_r N_r$, and $\rho \in \{0.01, 0.05, 0.1\}$ specifies the target tail-to-head ratio. Smaller ρ produces stronger imbalance, with the realized ratio subject to rounding and available class counts. We repeat sampling to match the full-data sample count per epoch. Keeping the number of epochs and evaluation splits unchanged.

B.1.2 Parameter Scaling and Model Selection

Let $\Delta\boldsymbol{\theta}$ denote the learned parameter change from $\boldsymbol{\theta}_0$. After training, we evaluate $\boldsymbol{\theta}_0 + \lambda\Delta\boldsymbol{\theta}$ for methods trained directly for forgetting, including FR, and $\boldsymbol{\theta}_0 - \lambda\Delta\boldsymbol{\theta}$ for TA, OrthoReg, LTA, NM, and TS. All methods share the scale grid $\lambda \in \{0.05, 0.10, \dots, 1.50\}$. We select the configuration with the lowest forget validation accuracy subject to preserving at least 95% of pretrained accuracy on the ImageNet retain validation subset. The selected configuration is then evaluated on the test splits.

B.1.3 Vision Baselines

Random Labeling (RL). We assign a fixed random label to each forget-training example and minimize cross-entropy on these labels.

Entropy Maximization (EM). We maximize the entropy of class predictions on forget data.

Gradient Ascent (GA). We maximize cross-entropy on forget data using the original labels.

SalUn. Our adaptation of SalUn (Fan et al., 2024) selects parameters through forget-gradient saliency and applies gradient ascent to the selected parameters. We search the selected fraction $q \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$ using full-data ViT-Base/32 validation results and reuse the chosen fraction across vision experiments.

Task Arithmetic (TA). TA (Ilharco et al., 2023) minimizes the forget-data classification loss and negates the resulting task vector. Its negation scale is selected using the validation protocol above.

OrthoReg (OR). OrthoReg (Liu et al., 2026) adds orthogonality regularization to task-vector fitting. We search its regularization coefficient $\alpha_{\text{OR}} \in \{0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10, 15\}$ using the same validation criterion.

Task Simplex (TS). Following Task Simplex (Dong et al., 2025), we construct 30 task vectors from separate fine-tuning runs with different RandAugment (Cubuk et al., 2020) configurations and combine predictions using the task-simplex rule.

NegMerge. We apply NegMerge (Kim et al., 2025) to the same 30 task vectors, merging the components with consistent signs across task vectors before negation.

Linearized Task Arithmetic (LTA). We follow the authors' implementation (Ortiz-Jimenez et al., 2023) of fine-tuning and task-vector negation in the model's tangent space.

B.1.4 Ablation Variants

Let $\Delta \boldsymbol{z} = \boldsymbol{z} - \boldsymbol{z}_0$ and $\boldsymbol{z}_r = 2\boldsymbol{z}_0 - \boldsymbol{z}$. Full FR minimizes the classification loss on $\boldsymbol{z}_0 - \Delta \boldsymbol{z} \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{1}, \sigma^2 \mathbf{I})$. We select hyperparameters for all variants using the same validation criterion.

FR with uniform noise. We replace the Gaussian multipliers with independent components $\boldsymbol{\epsilon} \sim \mathcal{U}(1-a, 1+a)$ and search the half-width $a \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$.

FR with dropout noise. We use $\boldsymbol{\epsilon} = \boldsymbol{b}/p$, where each $\boldsymbol{b} \sim \text{Bernoulli}(p)$ independently, and search the keep probability $p \in \{0.1, 0.2, 0.5, 0.8, 0.9, 0.95\}$.

FeatReg. FeatReg adds a feature-deviation penalty $\alpha_{\text{feat}} \|\boldsymbol{z} - \boldsymbol{z}_0\|_2^2$ to the training objective of each combined method. We search α_{feat} over {1e-4, 5e-4, 1e-3, 5e-3, 0.01, 0.05, 0.1, 0.2, 0.5, 1}.

ParamReg. ParamReg adds a parameter-deviation penalty $\alpha_{\text{param}} \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2$ to the training objective. We search α_{param} over {1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 0.01, 0.05, 0.1}.

B.2 Language Unlearning

B.2.1 Datasets and Training

We evaluate Zephyr-7B-β (Tunstall et al., 2024) separately on WMDP-Bio and WMDP-Cyber (Li et al., 2024), training on each domain's forget corpus. We randomly reserve 20% of each domain's multiple-choice questions for validation and the remaining 80% for testing. MMLU (Hendrycks et al., 2021) validation and test splits measure retention. All methods share these evaluation splits. Experiments use OpenUnlearning (Dorna et al., 2025) with batch size 32, default learning rate 1e-5, weight decay 1e-3, and linear learning-rate decay.

B.2.2 Evaluation and Model Selection

WMDP and MMLU use four answer choices, giving chance accuracy $c = 25\%$. Let A_f^0, A_r^0 denote pretrained forget and retain accuracies, and A_f, A_r their values after unlearning. We define the chance-normalized forgetting and retention scores as

$$S_f = \text{clip}\Big(\frac{A_f^0 - A_f}{A_f^0 - c}, 0, 1\Big), \qquad S_r = \text{clip}\Big(\frac{A_r - c}{A_r^0 - c}, 0, 1\Big).$$

Following OpenUnlearning (Dorna et al., 2025), we aggregate the two scores by their **Harmonic Mean**,

$$H = \frac{2 S_f S_r}{S_f + S_r},$$

with $H = 0$ when $S_f + S_r = 0$. We select hyperparameters separately for each domain by maximizing validation Harmonic Mean H.

B.2.3 Language Baselines

All methods use only forget data for parameter updates. Retain-data terms are omitted where present in the original objectives.

Gradient Ascent (GA). We maximize forget-corpus cross-entropy and search the learning rate η over {1e-6, 2e-6, 5e-6, 1e-5, 2e-5}.

Task Arithmetic (TA). We fit the forget corpus and negate the learned parameter change. We search the same learning-rate grid as GA and the post-training negation scale $\lambda \in \{0.1, 0.2, \ldots, 1.2\}$.

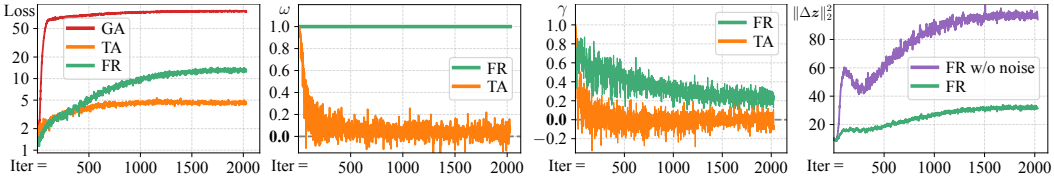
Negative Preference Optimization (NPO). NPO (Zhang et al., 2024) uses the pretrained model as a fixed reference. We search its inverse-temperature parameter β over {1e-3, 5e-3, 0.01, 0.05, 0.1, 0.5, 1, 2}.

SimNPO. SimNPO (Fan et al., 2025) normalizes its reward by response length. We search the loss-scaling parameter $\beta \in \{0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10, 15, 20\}$ and reward margin δ over {0, 1e-3, 5e-3, 0.01, 0.05, 0.1}.

Weighted Gradient Ascent (WGA). WGA (Wang et al., 2025) weights token-level ascent according to model confidence. We search its confidence-weighting exponent α_{WGA} over {1e-3, 5e-3, 1e-2, 5e-2, 0.1, 0.5}.

Representation Misdirection for Unlearning (RMU). We follow OpenUnlearning's implementation of RMU (Li et al., 2024) and search the steering coefficient $c_{\text{RMU}} \in \{0.1, 0.2, 0.5, 1, 2, 5, 10, 15, 20\}$.

C More Experiments



(a) Unlearning loss. (b) Jacobian alignment ω. (c) Ascent alignment γ. (d) Feature deviation.

Figure 6: Training dynamics on Stanford Cars with ViT-Base/32: (a) unlearning loss, (b) Jacobian alignment, (c) ascent alignment, and (d) feature deviation. See Figure 4 for SUN397 results.

D Storage Cost of the Feature Memory

Table 6 reports the storage cost of the feature memory, which holds the pretrained ViT-Base/32 features of each dataset. Averaged over the eight datasets, these features occupy only 52.35 MB. Such a memory requirement is negligible compared with the 6.64 GB of GPU memory required to train the backbone, amounting to only 0.77% of it. Owing to this small footprint, the entire feature memory can even be stored on the GPU.

Table 6: Storage (MB) of the pretrained ViT-Base/32 features on each dataset. The 52.35MB average is only 0.77% of the 6.64GB of GPU memory required to train the backbone.

Dataset	Stanford Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC45	SUN397	SVHN	Avg.
Storage	14.32	6.61	42.19	46.83	107.42	33.22	34.89	133.31	52.35