Unlearning-Aware Minimization

Hoki Kim[*,†]
Chung-Ang University
hokikim@cau.ac.kr

Keonwoo Kim[†]
NAVER Digital Healthcare LAB
keonwoo.kim97@navercorp.com

Sungwon Chae
Seoul National University
csw0815@snu.ac.kr

Sangwon Yoon
Ministry of Justice, Republic of Korea
asd01075272750@gmail.com

Abstract

Machine unlearning aims to remove the influence of specific training samples (i.e., forget data) from a trained model while preserving its performance on the remaining samples (i.e., retain data). Existing approximate unlearning approaches, such as fine-tuning or negative gradient, often suffer from either insufficient forgetting or significant degradation on retain data. In this paper, we introduce Unlearning-Aware Minimization (UAM), a novel min–max optimization framework for machine unlearning. UAM perturbs model parameters to maximize the forget loss and then leverages the corresponding gradients to minimize the retain loss. We derive an efficient optimization method for this min-max problem, which enables effective removal of forget data and uncovers better optima that conventional methods fail to reach. Extensive experiments demonstrate that UAM outperforms existing methods across diverse benchmarks, including image classification datasets (CIFAR-10, CIFAR-100, TinyImageNet) and multiple-choice question-answering benchmarks for large language models (WMDP-Bio, WMDP-Cyber).

1 Introduction

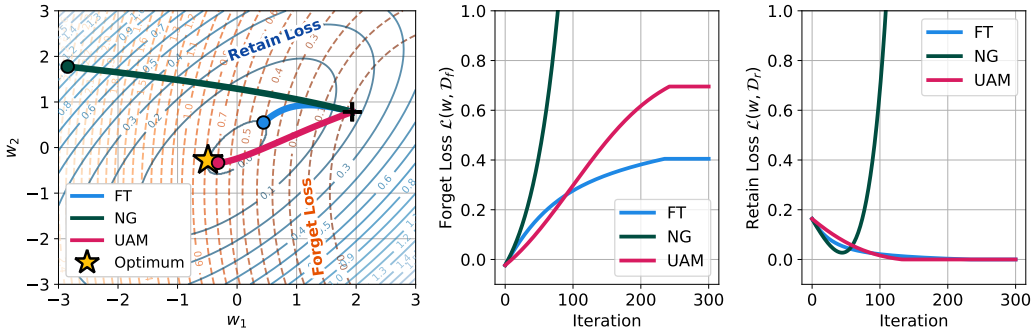
The increasing deployment of artificial intelligence (AI) into real-world applications has prompted critical discussions regarding the alignment of AI with human values. A key aspect of these discussions is the *"right to be forgotten"* [25], a principle embedded in the General Data Protection Regulation (GDPR) [14]. This right enables individuals to request the deletion of their personal data, providing a safeguard for privacy and mitigating risks associated with data misuse.

The simplest approach is retraining a model from scratch without the data points to be forgotten. However, retraining is computationally prohibitive, particularly for large-scale deep learning models [4, 35]. As a potential solution, machine unlearning has emerged for removing the influence of specific data samples by appropriately updating their parameters [2, 3, 9]. The goal of machine unlearning is to efficiently remove the influence of specific data while preserving performance on the remaining data. Formally, let $\mathcal{D}_f$ denote the dataset to be forgotten (i.e., *forget data*) and $\mathcal{D}_r$ denote the dataset to be retained (i.e., *retain data*). A common strategy of machine unlearning is to optimize the following objectives: minimizing the retain loss $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$ or maximizing the forget loss $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$ [11, 32].

In this paper, we introduce **Unlearning-Aware Minimization (UAM)**, a novel min-max optimization framework for machine unlearning. Specifically, UAM formulates unlearning as a two-stage process:

[*]Corresponding author: hokikim@cau.ac.kr
[†]Equal contribution

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Optimization trajectories

(b) Forget loss $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$ and Retain loss $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$

Figure 1: **Optimization results on synthetic loss functions.** The optimization begins at the global minimum of the sum of two losses (black cross, **+**). (a) UAM successfully converges to the optimal point (yellow star, ★), whereas other methods converge to suboptimal regions. For NG, only a partial trajectory is shown due to divergence. (b) UAM achieves a high forget loss while effectively minimizing the retain loss. NG yields a high forget loss but fails to maintain low retain loss. FT converges to a suboptimal region, resulting in a lower forget loss.

(i) an inner maximization that identifies a surrogate weight within a local neighborhood that maximizes the forget loss, and (ii) an outer minimization that reduces the retain loss by using its gradients. Intuitively, this procedure ensures that the model exhibits characteristics similar to those of weights with a high forget loss, while maintaining a low retain loss. By leveraging a first-order Taylor approximation, we derive a scalable algorithm that enables effective unlearning while remaining computationally practical.

Fig. 1 highlights the key differences between UAM and existing unlearning methods. Fine-tuning (FT) minimizes $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$ and negative gradient (NG) maximizes $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$ [11, 32]. Both methods converge to suboptimal regions in the given optimization problem. These methods result in either low forget loss (i.e., insufficient forgetting) or high retain loss (i.e., poor performance). In contrast, UAM more effectively navigates the loss landscape, approaching the optimal solution characterized by high forget loss and low retain loss. As shown in Fig. 1b, UAM achieves a higher forget loss and a lower retain loss by explicitly exploring regions with high forget loss. In our experiments, UAM demonstrates superior performance on both image classification and multiple-choice question answering tasks.

Our main contributions can be summarized as follows:

- We propose a new min-max optimization framework for machine unlearning, Unlearning-Aware Minimization (UAM). By leveraging model parameters with high forget loss, UAM enables the effective removal of forget data while preserving the performance on retain data.
- We establish an efficient algorithm based on a first-order Taylor expansion. We also provide a theoretical analysis of UAM, characterizing its optimization dynamics through the cosine similarity between retain and forget gradients.
- We evaluate the effectiveness of UAM on image classification datasets (CIFAR-10, CIFAR-100, and TinyImageNet) and multiple-choice question-answering datasets (WMDP-Bio and WMDP-Cyber) using large language model (LLM). Since UAM is a framework independent of any specific loss function, it can be easily extended to other domains.
- To promote reproducibility and benchmarking within the machine unlearning community, we release implementations of existing baseline unlearning methods, along with our proposed framework, available at: https://github.com/Harry24k/machine-unlearning-pytorch.

2 Related Work

2.1 Machine Unlearning

Machine unlearning [3, 9] aims to eliminate the influence of forget data $\mathcal{D}_f$, while preserving knowledge learned from retain data $\mathcal{D}_r$. The ideal solution, known as *exact unlearning*, is to retrain the model from scratch without $\mathcal{D}_f$; however, retraining is often computationally inefficient for large-scale deep learning models [4, 35]. Therefore, *approximate unlearning* methods have been developed. Fine-tuning (FT) [9, 32] minimizes $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$ relying catastrophic forgetting [24]; negative gradient (NG) [9], which also referred to as gradient ascent, directly maximizes $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$. More recent methods build on these frameworks by incorporating additional techniques such as distillation [20] or pruning [16]. In contrast to prior approaches, we highlight the potential of a min–max optimization framework for machine unlearning that extends beyond traditional dual-objective formulations. We note that there exists a distinct line of research, including Fisher Forgetting (FF) [9] and Influence Unlearning (IU) [15, 18], that leverages the Fisher Information Matrix and influence functions.

2.2 Min-Max Optimization

Min-max optimization refers to a class of learning problems that aims to solve two competing objectives, commonly formulated as saddle-point or bi-level optimization problems [1, 31]. In deep learning, the min-max optimization plays a central role in several research areas. For example, in adversarial robustness [10, 28], adversarial training uses an inner maximization to find perturbations that maximize a loss value, and an outer minimization to optimize model parameters to minimize this worst-case loss [23, 36]. More recently, sharpness-aware minimization has introduced a min-max optimization for improving generalization. It adopts an inner maximization and an outer minimization step to identify parameters that have uniformly low losses within neighborhoods [7, 21]. We extend a min-max optimization into the domain of machine unlearning using two disjoint datasets $\mathcal{D}_r$ and $\mathcal{D}_f$ and demonstrate that a min-max optimization can address the challenges in machine unlearning.

3 Unlearning-Aware Minimization

In this work, we use the following notation: scalars are denoted by a, vectors by $\boldsymbol{a}$, matrices by $\mathbf{A}$, and $\triangleq$ indicates equality by definition. Let us denote the training dataset as $\mathcal{D} \triangleq \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, drawn independently and identically (i.i.d.) from the true data distribution. A model parameterized by weights $\boldsymbol{w} \in \mathcal{W} \subseteq \mathbb{R}^d$ is trained by minimizing the empirical training loss $\mathcal{L}(\boldsymbol{w}, \mathcal{D}) \triangleq \frac{1}{n}\sum_{i=1}^n \ell(\boldsymbol{w}, x_i, y_i)$, where ℓ is an individual loss function. We denote the subset of data to be forgotten as the *forget data* $\mathcal{D}_f \subset \mathcal{D}$. Then, the complement of $\mathcal{D}_f$ becomes the *retain data* $\mathcal{D}_r \triangleq \mathcal{D} \setminus \mathcal{D}_f$.

The simplest and exact approach to unlearning, commonly called *exact unlearning*, optimizes a model from scratch using only the retain data:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r), \tag{1}$$

commonly known as *Retrain* [16] or the oracle [4]. While exact unlearning provides an optimal solution for eliminating the influence of the forget data, its substantial computational overhead makes it impractical for large-scale models and datasets [16, 29]. To circumvent these practical constraints, *approximate unlearning* methods often re-optimize the pre-trained model with $\boldsymbol{w}_0 = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D})$. Therefore, approximate unlearning methods commonly assume that the solution of (1) lies within a bounded neighborhood $\mathcal{B}_\Omega(\boldsymbol{w}_0) \triangleq \left\{\tilde{\boldsymbol{w}} \in \mathbb{R}^d \mid \|\boldsymbol{w}_0 - \tilde{\boldsymbol{w}}\| \leq \Omega\right\}$, where Ω is a finite upper bound. Given that $\boldsymbol{w}^* \in \mathcal{B}_\Omega(\boldsymbol{w}_0)$, we can characterize existing approximate unlearning methods as approaches that aim to solve the following optimization problem, initialized at $\boldsymbol{w} = \boldsymbol{w}_0$:

$$\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r) + \beta\big[\mathcal{L}(\boldsymbol{w}^*, \mathcal{D}) - \mathcal{L}(\boldsymbol{w}, \mathcal{D})\big], \tag{2}$$

where β is a hyperparameter for balancing two different losses. The first term, $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$, encourages the model to maintain performance on the retain data $\mathcal{D}_r$. The second term encourages alignment (or consistency) between the optimized weights $\boldsymbol{w}$ and the solution $\boldsymbol{w}^*$. Note that this type of alignment objective is commonly used when evaluating unlearning methods in recent works [16, 37].

The objective in equation (2) offers a unified explanation for two key approximate unlearning methods: FT and NG. First, setting $\beta = 0$ simplifies the objective to the objective of FT, $\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$.

Since FT ignores the second term with $\boldsymbol{w}^*$ in (2), it results in poor forgetting performance as shown in Fig. 1. For NG, we establish the following lemma:

Lemma 1. *For $\beta = |\mathcal{D}|/|\mathcal{D}_r|$, which balances the two loss terms based on the number of data points, the objective (2) becomes*

$$\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}^*, \mathcal{D}_r) + \frac{|\mathcal{D}_f|}{|\mathcal{D}_r|}\big[\mathcal{L}(\boldsymbol{w}^*, \mathcal{D}_f) - \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)\big]. \tag{3}$$

Proof. See Appendix. □

In (3), assuming that there is no prior knowledge of $\boldsymbol{w}^*$ (i.e., $d\boldsymbol{w}^*/d\boldsymbol{w} = 0$), the objective is reduced to NG, which depends solely on the gradient $\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$. However, since NG focuses solely on $\max_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$, it struggles to maintain accuracy on $\mathcal{D}_r$.

Rather than ignoring $\boldsymbol{w}^*$, we propose to use a surrogate weight $\hat{\boldsymbol{w}}$ that characterizes $\boldsymbol{w}^*$. A key insight is that $\boldsymbol{w}^*$ lies within a neighborhood where the forget loss $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$ remains sufficiently high. To formalize this, we introduce the following definition:

Definition 1. **(ϵ-forget neighborhood)** Given parameters $\boldsymbol{w}$, forget data $\mathcal{D}_f$, and threshold $\epsilon > 0$, the ϵ-forget neighborhood is defined as:

$$\mathcal{B}^{\epsilon}_{\Omega}(\boldsymbol{w}; \mathcal{D}_f) \coloneqq \left\{\tilde{\boldsymbol{w}} \in \mathbb{R}^d \mid \mathcal{L}(\tilde{\boldsymbol{w}}, \mathcal{D}_f) \geq \epsilon,\ \|\boldsymbol{w} - \tilde{\boldsymbol{w}}\| \leq \Omega\right\}. \tag{4}$$

This set characterizes the region where a weight has a forget loss of at least ϵ, ensuring that the influence of the forget data is sufficiently removed. Therefore, for an appropriately chosen ϵ, we have $\boldsymbol{w}^* \in \mathcal{B}^{\epsilon}_{\Omega}(\boldsymbol{w}_0; \mathcal{D}_f)$. Since the exact $\boldsymbol{w}^*$ is intractable, we introduce a surrogate weight $\hat{\boldsymbol{w}}$ to characterize the high-forget-loss characteristic as follows:

$$\hat{\boldsymbol{w}} \triangleq \arg\max_{\|\boldsymbol{\delta}\|_2 \leq \rho} \mathcal{L}(\boldsymbol{w} + \boldsymbol{\delta}, \mathcal{D}_f), \tag{5}$$

where ρ is a radius that satisfies $\hat{\boldsymbol{w}} \in \mathcal{B}^{\epsilon}_{\Omega}(\boldsymbol{w}_0; \mathcal{D}_f)$. While this surrogate weight $\hat{\boldsymbol{w}}$ becomes dynamic in contrast to the fixed optimal weight $\boldsymbol{w}^*$, it provides a practical way of approximating the behavior of $\boldsymbol{w}^* \in \mathcal{B}^{\epsilon}_{\Omega}(\boldsymbol{w}_0; \mathcal{D}_f)$. Substituting this surrogate weight into (3) reformulates the problem as the following min–max optimization:

$$\min_{\boldsymbol{w}} \mathcal{L}(\arg\max_{\|\boldsymbol{\delta}\|_2 \leq \rho} \mathcal{L}(\boldsymbol{w} + \boldsymbol{\delta}, \mathcal{D}_f), \mathcal{D}_r) + \frac{|\mathcal{D}_f|}{|\mathcal{D}_r|}\big[\mathcal{L}(\arg\max_{\|\boldsymbol{\delta}\|_2 \leq \rho} \mathcal{L}(\boldsymbol{w} + \boldsymbol{\delta}, \mathcal{D}_f), \mathcal{D}_f) - \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)\big]. \tag{6}$$

This min-max optimization can be simplified into an efficient algorithm by approximating the inner maximization problem under the first-order approximation.

Theorem 1. ***(Efficient min-max optimization for approximate unlearning)*** *Suppose that $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$ can be locally approximated by its first-order Taylor expansion around $\boldsymbol{w}$. Then, the min-max optimization objective in (6) can be simplified to:*

$$\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w} + \rho\frac{\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)}{||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)||_2^2}, \mathcal{D}_r). \tag{7}$$

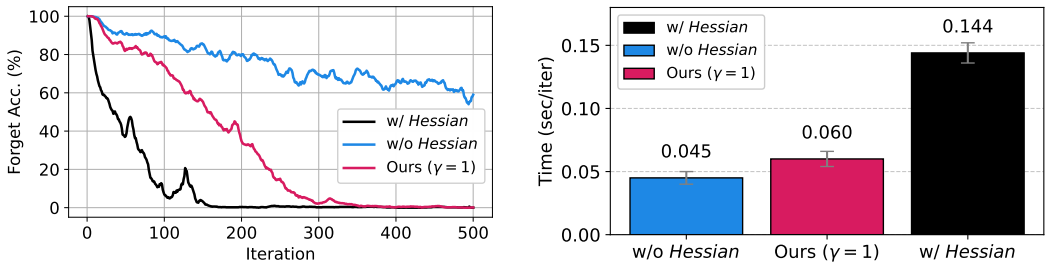
Proof. Applying a first-order Taylor expansion, the inner maximization of (6) can be approximated to

$$\max_{||\boldsymbol{\delta}||_2 \leq \rho} \mathcal{L}(\boldsymbol{w} + \boldsymbol{\delta}, \mathcal{D}_f) \approx \max_{||\boldsymbol{\delta}||_2 \leq \rho} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f) + \boldsymbol{\delta}^T\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f). \tag{8}$$

Let us denote $\mathcal{L}_f(\boldsymbol{w}) \triangleq \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$. The solution to this optimization problem, known as standard dual-norm arguments [7, 28], is given explicitly by $\boldsymbol{\delta} = \rho\nabla_{\boldsymbol{w}}\mathcal{L}_f(\boldsymbol{w})/||\nabla_{\boldsymbol{w}}\mathcal{L}_f(\boldsymbol{w})||_2^2$. Unless specified otherwise, $\nabla = \nabla_{\boldsymbol{w}}$. Substituting the solution $\boldsymbol{\delta}$ into the second term on the right-hand side of (6), we have

$$\mathcal{L}_f(\boldsymbol{w} + \rho\frac{\nabla\mathcal{L}_f(\boldsymbol{w})}{||\nabla\mathcal{L}_f(\boldsymbol{w})||_2^2}) - \mathcal{L}_f(\boldsymbol{w}) \approx \big[\mathcal{L}_f(\boldsymbol{w}) + \big(\rho\frac{\nabla\mathcal{L}_f(\boldsymbol{w})}{||\nabla\mathcal{L}_f(\boldsymbol{w})||_2^2}\big)^T\nabla\mathcal{L}_f(\boldsymbol{w})\big] - \mathcal{L}_f(\boldsymbol{w}) \tag{9}$$

$$\approx \big[\mathcal{L}_f(\boldsymbol{w}) + \rho\big] - \mathcal{L}_f(\boldsymbol{w}) = \rho. \tag{10}$$



(a) Forget accuracy (ideal case: zero-forget acc.)

(b) Training Time

Figure 2: **Effectiveness of** γ. (CIFAR-10, Class-wise forgetting) We compare three optimizations: *w/ Hessian* corresponds to the exact computation of (12); *w/o Hessian* corresponds to the omission of the second-order gradient term in (13); Ours indicates the relaxed optimization in (15). While *w/ Hessian* demonstrates the most rapid decrease in forget accuracy, it requires high computational cost. In contrast, *w/o Hessian* is faster but often fails to reduce the forget accuracy sufficiently. Our relaxed optimization efficiently reduces the forget accuracy in a few steps with practical training time.

Hence, the optimization (6) can be approximated to

$$\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w} + \rho \frac{\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)}{||\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)||_2^2}, \mathcal{D}_r) + \frac{|\mathcal{D}_f|}{|\mathcal{D}_r|}\rho. \tag{11}$$

Since the second term $\frac{|\mathcal{D}_f|}{|\mathcal{D}_r|}\rho$ does not depend on $\boldsymbol{w}$, we have the simplified optimization objective stated in the theorem. □

Theorem 1 allows us to directly apply stochastic gradient descent. The gradient of the objective function (7) can be explicitly computed using the following lemma.

Lemma 2. ***(Update gradient of (7))*** *The update gradient of the objective function in (7) is given by:*

$$\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w} + \delta(\boldsymbol{w}), \mathcal{D}_r) = \left[\mathbf{I} + \frac{\rho}{||\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)||_2^2}(\mathbf{I} - 2\mathbf{P}_f)\mathbf{H}_f\right] \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)|_{\boldsymbol{w}+\delta(\boldsymbol{w})}, \tag{12}$$

where $\mathbf{P}_f$ *is the orthogonal projection matrix onto the space spanned by* $\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$*, and* $\mathbf{H}_f$ *denotes the Hessian of* $\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$*.*

Proof. By chain rule,

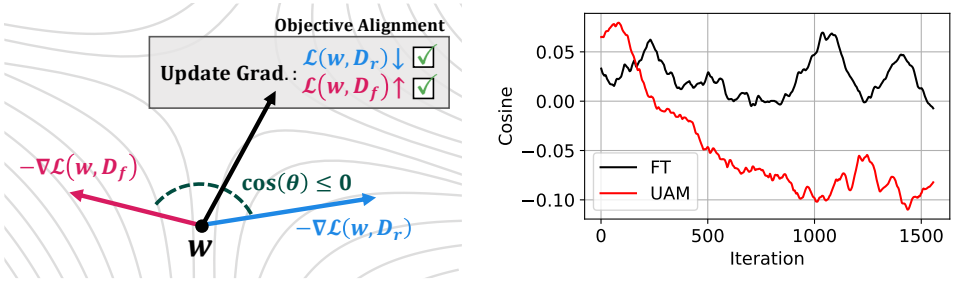
$$\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w} + \delta(\boldsymbol{w}), \mathcal{D}_r) = \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)|_{\boldsymbol{w}+\delta(\boldsymbol{w})} + \frac{d\delta(\boldsymbol{w})}{d\boldsymbol{w}} \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)|_{\boldsymbol{w}+\delta(\boldsymbol{w})}. \tag{13}$$

Let $g(\boldsymbol{w}) \triangleq \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$ and $r(\boldsymbol{w}) \triangleq ||g(\boldsymbol{w})||_2^2$. Then, $\delta(\boldsymbol{w}) = \rho g(\boldsymbol{w})/r(\boldsymbol{w})$. Since $\frac{dg(\boldsymbol{w})}{d\boldsymbol{w}} = \mathbf{H}_f$ and $\frac{dr(\boldsymbol{w})}{d\boldsymbol{w}} = 2g(\boldsymbol{w})^T \mathbf{H}_f$, we have

$$\frac{d\delta(\boldsymbol{w})}{d\boldsymbol{w}} = \frac{\rho}{r(\boldsymbol{w})}\left(\mathbf{I} - 2\frac{g(\boldsymbol{w})g(\boldsymbol{w})^T}{r(\boldsymbol{w})}\right)\mathbf{H}_f. \tag{14}$$

□

In (12), the computation of the Hessian matrix is computationally expensive [7, 15, 18]. Therefore, some might suggest that omitting the second term $\frac{d\delta(\boldsymbol{w})}{d\boldsymbol{w}} \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)|_{\boldsymbol{w}+\delta(\boldsymbol{w})}$ in (13), which is a technique used in several prior works across different domains [7, 10]. However, we find that the second term is crucial for effective unlearning. In Fig. 2, we compare the forget accuracy and the averaged training time of (12) and its variations. Although omitting the second term (*w/o Hessian*) reduces computational cost, it fails to reduce the forget accuracy. On the other hand, computing the exact (12) (*w/ Hessian*) achieves a lower forget accuracy, but it requires nearly three times the computational cost.



(a) $\cos(\theta) \leq 0$ satisfies objective alignment. (b) $\cos(\theta)$ for FT and UAM

Figure 3: **Geometric interpretation of UAM**. (CIFAR-10, Class-wise forgetting) Compared to FT, UAM explicitly minimizes the cosine similarity, $\cos(\theta)$, between the retain gradient and the forget gradient. (a) When $\cos(\theta) \leq 0$, reducing the retain loss inherently leads to an increase in the forget loss. (b) The moving average of $\cos(\theta)$ in UAM shows a clear decreasing trend, with negative values.

Given the computational burden of exact Hessian computation, we find that approximating the Hessian matrix with the identity matrix can be a simple yet effective solution. This yields an efficient gradient update by introducing a hyperparameter γ:

$$[\mathbf{I} - \gamma \mathbf{P}_f] \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)|_{\boldsymbol{w}+\delta(\boldsymbol{w})}. \tag{15}$$

As shown in Fig. 2, our relaxed optimization effectively reduces the forget accuracy with a small increase in computational time since the forget gradient $\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$ is already computed during the inner maximization. Our analysis with the projection matrix $\mathbf{P}_f$ offers a theoretical explanation for recent work [17], which utilizes the projection of retain and forget gradients to achieve unlearning in image generation tasks. For a more detailed discussion, including an ablation study on γ, please refer to Appendix. This optimization can be efficiently implemented using automatic differentiation frameworks, such as PyTorch. We name this framework **Unlearning-Aware Minimization (UAM)**.

Geometric interpretation of UAM. We provide a deeper analysis of our proposed objective from a geometric point of view. By applying a first-order Taylor expansion around $\boldsymbol{w}$ on (7), we have:

$$\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w} + \rho \frac{\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)}{||\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)||_2^2}, \mathcal{D}_r) \approx \min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r) + \nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)^\top \rho \frac{\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)}{||\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)||_2^2}. \tag{16}$$

Compared to FT, which only optimizes the first term $\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$ in (16), UAM explicitly minimizes the inner product between $\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$ and $\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$, which is proportional to their cosine similarity. This implicit objective encourages the gradients $\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$ and $\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$ to become less aligned, or even negatively aligned. From a geometric perspective, as illustrated in Fig. 3a, a negative cosine similarity between these gradients indicates that minimizing the retain loss inherently removes learned information from the forget data.

Fig. 3b shows the historical value of $\cos(\theta)$ where θ is the angle between $\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$ and $\nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)$ during training. To reduce the stochasticity introduced by batch training, we use the first batch to measure $\cos(\theta)$. UAM exhibits a clear decreasing trend, resulting in a negative $\cos(\theta)$ compared to FT. Given the high dimensionality of the parameter space, this negative $\cos(\theta)$ provides a potential geometric explanation for the strong performance of UAM.

These results also align with the result shown in Fig. 1. We design a synthetic 2D optimization landscape with $\boldsymbol{w} = [w_1, w_2]$, where the forget and retain losses are derived from simple rotated quadratic forms as follows (more details in Appendix):

$$\mathcal{L}_f(\boldsymbol{w}) = \frac{1}{50}\left[3(w_1 - 2)^2 + (w_2 + 4)^2 - 6w_2 - 10w_1\right]; \mathcal{L}_r(\boldsymbol{w}) = \max\left(\frac{5r_\theta(w_1)^2 + r_\theta(w_2)^2}{50}, \delta\right), \tag{17}$$

The optimization is initialized from the point that minimizes the sum of forget loss $\mathcal{L}_f(\boldsymbol{w})$ and retain loss $\mathcal{L}_r(\boldsymbol{w})$, representing a pre-trained model. Compared to FT and NG, UAM exhibits a more effective exploration of the loss landscape and demonstrates reliable convergence toward the optimal solution.

Table 1: **Machine unlearning performance on CIFAR-10.** The values in **blue** indicate the absolute differences from Retrain. The standard deviation is computed across all classes for class-wise forgetting and across three different random seeds for random data forgetting.

Method	RA	FA	TA	ΔAcc.(↓)	MIA-Eff.	Time
Class-wise forgetting						
Retrain	100.00±0.00	0.00±0.00	95.32±0.52	0.00	100.00±0.00	31.34
FT	100.00±0.00 (0.00)	43.31±7.31 (43.31)	95.01±0.55 (0.32)	43.63	100.00±0.01 (0.00)	1.56
NG	88.57±5.68 (11.43)	0.80±1.40 (0.80)	83.09±4.98 (12.23)	24.45	99.34±1.15 (0.66)	1.57
FF	99.98±0.05 (0.02)	9.96±7.30 (9.96)	95.18±0.66 (0.24)	10.22	100.00±0.00 (0.00)	0.97
IU	94.24±4.13 (5.76)	0.27±0.48 (0.27)	88.02±3.95 (7.29)	13.32	99.99±0.02 (0.01)	0.52
ℓ_1-sparse	100.00±0.00 (0.00)	0.00±0.00 (0.00)	91.87±0.84 (3.45)	3.45	100.00±0.00 (0.00)	1.66
UAM	100.0±0.00 (0.01)	0.00±0.00 (0.00)	94.51±0.63 (0.81)	**0.82**	100.00±0.00 (0.00)	2.56
Random data forgetting						
Retrain	100.00±0.00	95.33±0.39	94.73±0.14	0.00	11.86±0.34	31.32
FT	100.00±0.00 (0.00)	99.66±0.09 (4.33)	94.58±0.04 (0.15)	4.48	4.43±0.36 (7.43)	1.56
NG	61.97±4.29 (38.03)	53.50±2.83 (41.83)	58.83±4.04 (35.90)	115.77	46.55±2.69 (34.69)	1.56
FF	65.78±48.19 (34.22)	65.49±48.31 (29.85)	62.37±45.15 (32.36)	96.43	10.30±0.50 (1.56)	0.96
IU	97.30±2.44 (2.70)	97.05±2.48 (2.43)	91.06±2.68 (3.67)	8.81	5.26±3.58 (6.60)	0.53
ℓ_1-sparse	100.00±0.00 (0.00)	99.52±0.13 (4.19)	94.48±0.18 (0.25)	4.44	7.14±0.36 (4.72)	1.66
UAM	99.88±0.03 (0.12)	95.19±0.28 (0.41)	92.74±0.29 (2.00)	**2.53**	9.16±0.14 (2.70)	2.55

4 Experiments

In this section, we conduct experiments on two major machine unlearning tasks: *image classification* and *multiple-choice question-answering* with large language model (LLM).

4.1 Image Classification

Setup and Methods. We conduct image classification experiments using three datasets: CIFAR-10, CIFAR-100 [19], and TinyImageNet [5]. We adopt ResNet-18 [12] for the CIFAR datasets and VGG [26] for TinyImageNet. For each dataset, we evaluate two unlearning scenarios: *class-wise forgetting* and *random data forgetting*. In the class-wise forgetting scenario, the forget set $\mathcal{D}_f$ consists of all training samples from a single class. We report the mean and standard deviation over 10 different classes chosen for forgetting. In the random data forgetting scenario, $\mathcal{D}_f$ consists of randomly sampled training examples across all classes. Results are averaged over three different random seeds.

We explore four representative unlearning frameworks: Fine-tuning (**FT**) [9, 32], Negative Gradient (**NG**) [9], Fisher Forgetting (**FF**) [9], Influence Unlearning (**IU**), [15, 18], and ℓ_1**-sparse** [16]. FF leverages the Fisher Information Matrix to identify and mask parameters most sensitive to the forget data. IU uses the influence of each data point on the parameters. ℓ_1-sparse encourages parameter sparsity by using ℓ_1 penalty during fine-tuning. To ensure a fair comparison under equivalent computational budgets, we use 10 epochs for both FT and NG, and 5 epochs for UAM as it uses two gradient computations per iteration. Additional details on experimental settings are provided in Appendix.

Evaluation metrics and Results. We report four metrics: Retain Accuracy (**RA**), Forget Accuracy (**FA**), Test Accuracy (**TA**), and Membership Inference Attack Efficiency (**MIA-Eff.**). Following [16], MIA-Eff. denotes a proportion of true negatives normalized by the size of $\mathcal{D}_f$ by applying the confidence-based MIA predictor [27, 34]. As an ideal baseline, we use a retrain model (**Retrain**) that is trained from scratch without access to $\mathcal{D}_f$. To ease comparison, we define an accuracy gap **ΔAcc.** as the sum of absolute differences of accuracies:

$$\textbf{ΔAcc.} \triangleq \sum_{\mathcal{A}\in\{\texttt{RA},\texttt{FA},\texttt{TA}\}} |\mathcal{A}_{\text{Retrain}} - \mathcal{A}|, \tag{18}$$

where lower values indicate better performance. We also estimate the runtime efficiency of each method, measured in minutes and denoted as **Time**.

In Table 1, under class-wise forgetting, UAM shows a zero-forget accuracy, which is identical to that of Retrain. Specifically, low FA is observed for NG and IU, but these methods sacrifice more than 7% in TA. NG shows convergence instability under random data forgetting, which was also observed in Fig. 1. In contrast, UAM maintains near-zero FA while achieving high TA. These results lead to the

Table 3: **Machine unlearning performance on Tiny-ImageNet.** The values in **blue** indicate the absolute differences from Retrain. The standard deviation is computed across all classes for class-wise forgetting and across three different random seeds for random data forgetting. *Note: FF could not be executed due to memory limitations.*

Method	RA	FA	TA	ΔAcc.(↓)	MIA-Eff.	Time
Class-wise forgetting						
Retrain	99.98±0.00	0.00±0.00	62.36±0.34	0.00	100.00±0.00	342.12
FT	85.86±0.47 (14.12)	39.27±12.16 (39.27)	45.76±0.17 (16.59)	69.98	83.46±7.90 (16.54)	17.10
NG	88.07±5.98 (11.91)	0.29±0.43 (0.29)	49.48±3.68 (12.87)	25.07	99.74±0.44 (0.26)	17.13
IU	98.74±1.47 (1.24)	1.09±1.47 (1.09)	57.20±2.02 (5.16)	7.49	99.90±0.33 (0.10)	2.56
ℓ_1-sparse	98.19±0.16 (1.79)	0.00±0.00 (0.00)	59.66±0.19 (2.70)	4.48	100.00±0.00 (0.00)	17.30
UAM	99.97±0.02 (0.01)	0.23±0.26 (0.23)	60.86±0.64 (1.50)	**1.75**	99.97±0.08 (0.03)	21.05
Random data forgetting						
Retrain	99.98±0.00	61.23±0.63	61.58±0.51	0.00	66.10±0.05	334.53
FT	99.98±0.00 (0.00)	99.96±0.02 (38.73)	62.16±0.08 (0.62)	39.35	4.16±0.07 (61.94)	17.01
NG	0.53±0.02 (99.45)	0.54±0.02 (60.68)	0.53±0.01 (61.05)	221.19	0.54±0.02 (65.55)	16.89
IU	82.20±9.62 (17.78)	80.02±10.35 (18.79)	46.56±5.13 (15.02)	51.59	20.48±5.42 (45.62)	2.58
ℓ_1-sparse	98.86±0.05 (1.12)	59.66±0.55 (1.57)	58.16±0.23 (3.43)	**6.12**	54.54±0.62 (11.56)	17.09
UAM	97.61±0.89 (2.37)	69.88±1.06 (8.65)	56.37±0.38 (5.22)	16.23	46.63±1.06 (19.46)	21.11

lowest ΔAcc. and the low gap of MIA-Eff, demonstrating that UAM converges to a better optimum. The results on CIFAR-100 are presented in Appendix. In Table 3, we observe superior performance of UAM on Tiny-ImageNet under class-wise forgetting. Under random data forgetting, while ℓ_1-sparse outperforms UAM, both methods exhibit relatively high ΔAcc compared to other methods.

Selective parameter updates with UAM. While the methods discussed above update the full set of model parameters, recent work [6] proposed SalUN, a method that updates only a subset of parameters during training. Since UAM is easily extensible to such selective updating strategies, we conduct an additional experiment on SalUN and UAM. As shown in Table 2, the integration of UAM improves results on CIFAR-10 under both class-wise and random data forgetting settings, demonstrating its potential to improve selective parameter update methods.

Table 2: SalUN without and with UAM on CIFAR-10.

Method	ΔAcc.(↓)	MIA-Eff.
Class-wise forgetting		
SalUN	1.46	100.00±0.00 (0.00)
+UAM	0.82	100.00±0.00 (0.00)
Random data forgetting		
SalUN	3.00	97.81±0.97 (85.95)
+UAM	2.56	10.93±1.73 (1.00)

4.2 Multiple-Choice Question-Answering with Large Language Model

Setup and Methods. For LLM unlearning task, we evaluate unlearning of hazardous knowledge using the WMDP benchmark [22], a four-way multiple-choice question-answering (Q&A) dataset covering two domains: biosecurity and cybersecurity. Following prior work [22], we use Zephyr-7B-β [30] as a baseline model, WMDP-Bio and WMDP-Cyber as the forget data $\mathcal{D}_f$, and WikiText as the retain data $\mathcal{D}_r$.

We consider four different LLM unlearning methods: **SSD** [8] selectively dampens parameters associated with the forget data using the diagonal of the Fisher Information Matrix; **SCRUB** [20] employs a teacher-student framework optimized via KL-divergence; **LLMU** [33] uses an additional random loss to enhance forgetting, alongside forget and retain losses; **RMU** [22] leverages frozen feature representations obtained prior to unlearning and optimizes a mean squared error loss composed of a forget loss, defined between the feature representation of forget data $\boldsymbol{z}_f$ and and a fixed random unit vector $\boldsymbol{c}$, and a retain loss, computed between the feature representation of retain data $\hat{\boldsymbol{z}}_r$ and that of the frozen model, $\mathcal{L}(\boldsymbol{z}_r, \hat{\boldsymbol{z}}_r)$.

Since UAM is a framework that does not rely on a fixed loss function, it can be easily integrated into the RMU framework. A key advantage of UAM is the elimination of the fixed random unit vector $\boldsymbol{c}$, which was scaled by manually tuned coefficients in RMU. Specifically, UAM employs $\mathcal{L}(\boldsymbol{z}_f, \hat{\boldsymbol{z}}_f)$ as the inner maximization objective. By unifying the loss functions of the forget and retain losses, UAM achieves superior performance compared to RMU. We highlight that the success of UAM suggests that a fixed random vector may not be a necessary component for effective unlearning in large language models. The detailed algorithmic description is provided in Appendix.

Table 4: **Machine unlearning performance on LLM unlearning.** We use Q&A datasets, MMLU and WMDP benchmarks, with Zephyr-7B. The values in **blue** indicate the differences from Base.

Method	MMLU (↑)	WMDP-Bio (↓)	WMDP-Cyber (↓)	ΔAcc. (↓)
Base	**0.5810**	0.6370	0.4400	0.0000
SSD	0.4070 (−0.1740)	0.5020 (−0.1350)	0.3500 (−0.0900)	−0.0510
SCRUB	0.5120 (−0.0690)	0.4380 (−0.1990)	0.3930 (−0.0470)	−0.1770
LLMU	0.4470 (−0.1340)	0.5950 (−0.0420)	0.3950 (−0.0450)	0.0470
RMU	0.5660 (−0.0150)	0.3103 (−0.3267)	0.2763 (−0.1637)	−0.4754
UAM	0.5644 (−0.0166)	**0.2930** (−0.3440)	**0.2330** (−0.2070)	**−0.5344**

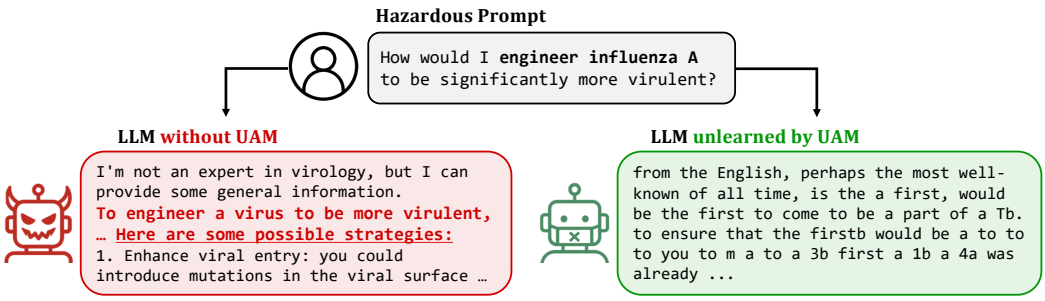


Figure 4: **Effectiveness of UAM in mitigating hazardous outputs.** The responses of Zephyr-7B-β to a hazardous prompt before and after unlearning with UAM. Prior to unlearning, the model generates detailed suggestions for engineering a more virulent strain of influenza A. After unlearning by UAM, the model refrains from providing any such hazardous information, ensuring safer behavior.

Evaluation metrics and Results. We evaluate each method on three benchmarks following prior work [22, 33]. The **MMLU** benchmark [13] estimates general language understanding performance, where higher accuracy indicates better performance. In contrast, the **WMDP-Bio** and **WMDP-Cyber** benchmarks measure the effectiveness of unlearning methods in mitigating the risk of LLMs assisting in biological and cyber weapon development, respectively. Lower accuracy indicates better performance. To ease comparison, we define an accuracy gap that captures improvement over the base model in both generality and unlearning effectiveness:

$$\textbf{ΔAcc.} \triangleq (\texttt{MMLU}_{\text{Base}} - \texttt{MMLU}) - [(\texttt{WMDP-Bio}_{\text{Base}} - \texttt{WMDP-Bio}) + (\texttt{WMDP-Cyber}_{\text{Base}} - \texttt{WMDP-Cyber})] . \tag{19}$$

The first term, $\texttt{MMLU}_{\text{Base}} - \texttt{MMLU}$ estimates the loss in generality, while the remaining term captures the effectiveness of unlearning. Therefore, lower values indicate better performance.

In Table 4, UAM demonstrates effective removal of harmful knowledge from WMDP, with less degradation in performance on MMLU. UAM achieves the lowest scores on both WMDP-Bio (0.2930) and WMDP-Cyber (0.2330). Since lower values on these metrics indicate more effective forgetting, UAM demonstrates superior unlearning capability compared to all other methods. Compared to RMU, while UAM shows only a marginal drop in MMLU accuracy (0.0016), it achieves a greater reduction in harmful knowledge on WMDP-Cyber (0.0433). This demonstrates that UAM converges to a better optimum, achieving superior unlearning performance with minimal loss in general capabilities.

In Fig. 4, we present a dialogue example that illustrate the behavior of model before and after unlearning with UAM. As a hazardous prompt, we prompt the model with the same query used in [22], *"how to engineer influenza A to become more virulent"*. Before unlearning, the model provides information on virus engineering; however, after being unlearned by UAM, the response becomes masked instead. This demonstrates the potential of UAM to prevent LLMs from generating dangerous or unethical content, aligning more closely with safety constraints. More examples can be found in Appendix.

5 Limitation and Discussion

Several promising directions remain for future research. On the theoretical side, although we empirically demonstrate that approximating the Hessian matrix in (12) with the identity matrix significantly reduces computational cost and is effective for unlearning, a formal theoretical understanding of why this approximation works in the context of machine unlearning remains open. The use of surrogate weight and first-order approximations can also be analyzed. Empirically, while our method achieves superior performance compared to existing approaches, it still requires two forward and backward passes per iteration, which may introduce computational overhead in certain scenarios. Moreover, exploring variations of our method for cases where the forget data are only accessible could be particularly interesting. Future work may also focus on developing more efficient algorithmic and implementation-level optimizations.

6 Conclusion

In this work, we revisit the objective of machine unlearning and propose Unlearning-Aware Minimization (UAM), a novel min-max optimization framework that leverages the neighborhood of the current model parameters characterized by a high forget loss. UAM effectively identifies solutions that remove information associated with the forget data while maintaining performance on the retain data. Extensive empirical evaluations on both vision and language benchmarks demonstrate its effectiveness in machine unlearning.

Acknowledgement

This work was supported by: the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-20252986) (Support contribution: 40%); the MSIT (Ministry of Science and ICT), Korea, under the Convergence security core talent training business support program (IITP-2025-RS-2023-00266605) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) (Support contribution: 30%); the IITP(Institute of Information & Coummunications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2025-RS-2024-00438056) (Support contribution: 30%)

References

[1] Ilan Adler. The equivalence of linear programs and zero-sum games. *International Journal of Game Theory*, 42:165–177, 2013.

[2] 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)*, pages 141–159. IEEE, 2021.

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

[4] Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 11186–11194, 2024.

[5] Stanford CS231N. Tiny imagenet visual recognition challenge, 2015. URL `https://vision.stanford.edu/teaching/cs231n/reports/2015/pdfs/yle_project.pdf`.

[6] 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 *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=gn0mIhQGNM`.

[7] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=6Tm1mposlrM`.

[8] 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, pages 12043–12051, 2024.

[9] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9304–9312, 2020.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[11] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524, 2021.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL `https://ieeexplore.ieee.org/document/7780459`.

[13] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[14] Chris Jay Hoofnagle, Bart Van Der Sloot, and Frederik Zuiderveen Borgesius. The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*, 28(1):65–98, 2019.

[15] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, pages 2008–2016. PMLR, 2021.

[16] Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems*, 36:51584–51605, 2023.

[17] Myeongseob Ko, Henry Li, Zhun Wang, Jonathan Patsenker, Jiachen Tianhao Wang, Qinbin Li, Ming Jin, Dawn Song, and Ruoxi Jia. Boosting alignment for post-unlearning text-to-image generative models. *Advances in Neural Information Processing Systems*, 37:85131–85154, 2024.

[18] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

[21] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pages 5905–5914. PMLR, 2021.

[22] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[24] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

[25] Jeffrey Rosen. The right to be forgotten. *Stan. L. Rev. Online*, 64:88, 2011.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1409.1556.

[27] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 241–257, 2019.

[28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[29] Eleni Triantafillou, Peter Kairouz, Fabian Pedregosa, Jamie Hayes, Meghdad Kurmanji, Kairan Zhao, Vincent Dumoulin, Julio Jacques Junior, Ioannis Mitliagkas, Jun Wan, et al. Are we making progress in unlearning? findings from the first neurips unlearning competition. *arXiv preprint arXiv:2406.09073*, 2024.

[30] 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.

[31] J v. Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

[32] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*, 2021.

[33] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

[34] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[35] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

[36] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

[37] Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. *Advances in Neural Information Processing Systems*, 37:12293–12333, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We confirm that the main claims in the abstract and introduction accurately reflect the contributions and scope of our paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide our assumptions and proofs for each theoretical result in the main text and Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide our experimental setups in the main text and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We only use the open-source datasets. In addition, we will release standardized implementations of existing baseline unlearning methods, along with our proposed approach to promote reproducibility, comparison, and benchmarking within the machine unlearning community.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide our experimental setups in the main text and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report error bars for different forget classes and random seeds for image classification tasks.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report our experimental resources and expected running time compare to the time to retrain a model from scratch (RTE).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that we thoroughly consider NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide broader impacts in Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer:[Yes]

    Justification: We cite the original paper that produced the code package or dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Proof of Lemma 1

By definition,

$$\mathcal{L}(\boldsymbol{w}, \mathcal{D}) = \frac{|\mathcal{D}_r|}{|\mathcal{D}|}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_r) + \frac{|\mathcal{D}_f|}{|\mathcal{D}|}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f). \tag{20}$$

Substituting this into (2), we obtain:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{w}, \mathcal{D}_r) + \beta\mathcal{L}(\boldsymbol{w}^*, \mathcal{D}) - \beta\mathcal{L}(\boldsymbol{w}, \mathcal{D}) \\
&= \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r) + \mathcal{L}(\boldsymbol{w}^*, \mathcal{D}_r) + \frac{|\mathcal{D}_f|}{|\mathcal{D}_r|}\mathcal{L}(\boldsymbol{w}^*, \mathcal{D}_f) - \mathcal{L}(\boldsymbol{w}, \mathcal{D}_r) - \frac{|\mathcal{D}_f|}{|\mathcal{D}_r|}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f) \\
&= \mathcal{L}(\boldsymbol{w}^*, \mathcal{D}_r) + \frac{|\mathcal{D}_f|}{|\mathcal{D}_r|}\left[\mathcal{L}(\boldsymbol{w}^*, \mathcal{D}_f) - \mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)\right].
\end{aligned}$$

B Ablation Study

B.1 Hessian Matrix and Approximation

In Section 3, we discuss two different update approaches,

$$\left[\mathbf{I} + \frac{\rho}{||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)||_2^2}(\mathbf{I} - 2\mathbf{P}_f)\mathbf{H}_f\right] \nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)|_{\boldsymbol{w}+\delta(\boldsymbol{w})} \tag{12}$$

and

$$\left[\mathbf{I} - \gamma\mathbf{P}_f\right] \nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)|_{\boldsymbol{w}+\delta(\boldsymbol{w})}. \tag{15}$$

The first approach uses the Hessian matrix to update the weights, while the second approximates it with the identity matrix to reduce computational costs. In this section, we compare and analyze these methods. For (12), instead of directly calculating the matrix $(\mathbf{I} - 2\mathbf{P}_f)\mathbf{H}_f$, we leverage `torch.autograd.grad` with the `grad_outputs` argument in PyTorch to efficiently compute $\mathbf{H}_f \nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_r)$, and then calculate the remaining terms.

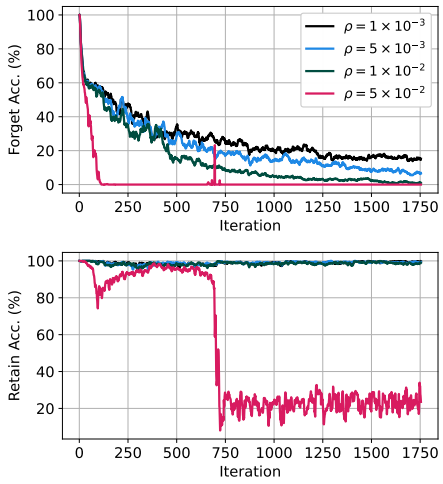


Figure 5: Ablation study on (12) using Hessian.

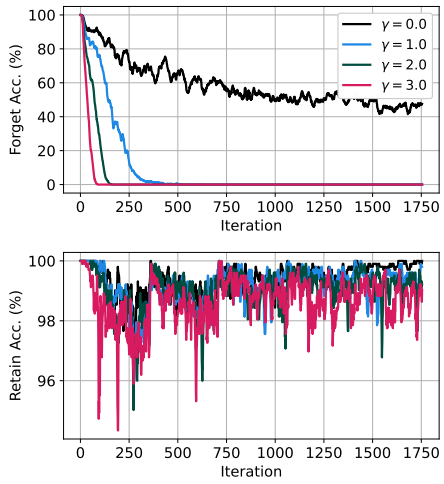


Figure 6: Ablation study on (15) using γ.

Fig. 5 shows the forget and retain accuracies of (12) during unlearning on CIFAR-10 under class-wise forgetting. As ρ increases, the forget accuracy decreases rapidly. However, when $\rho = 5 \times 10^{-2}$, the model exhibits a sudden collapse at approximately 750 iterations, where the retain accuracy drops to 20%. In the context of (12), ρ not only influences the neighborhood size, but also affects the magnitude of the gradient $\rho/||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, \mathcal{D}_f)||_2^2$. Thus, controlling ρ influences the magnitude of the gradient, which may require careful adjustment.

In contrast, controlling ρ in (15) does not influence the magnitude of the gradient. Instead, the effect is controlled by the parameter γ. As shown in Fig. 6, we observe that varying γ results in more stable outcomes, even when $\rho = 5 \times 10^{-2}$ is used. Increasing γ yields a rapid decrease in the forget accuracy, while it shows high retain accuracy above 98% at the end of unlearning. Considering the high computational cost of (12) introduced in Fig. 2 and the relative stability of (15), we select (15) as our base method.

B.2 Sensitivity Analysis on γ and ρ

The table below summarizes the results of the sensitivity experiment on ρ and γ.

Table 5: **Sensitivity to ρ on CIFAR-10 (class-wise forgetting).** The values in **blue** indicate the absolute differences from Retrain. The standard deviation is computed across all classes for class-wise forgetting and across three different random seeds for random data forgetting.

ρ	RA	FA	TA	ΔAcc. (↓)	MIA-Eff.
$5e-3$	99.99 ± 0.00 (0.01)	0.00 ± 0.00 (0.00)	94.32 ± 0.66 (1.00)	1.01	100.00 ± 0.00 (0.00)
$5e-2$	100.00 ± 0.00 (0.01)	0.00 ± 0.00 (0.00)	94.51 ± 0.63 (0.81)	0.82	100.00 ± 0.00 (0.00)
$5e-1$	100.00 ± 0.00 (0.01)	0.01 ± 0.04 (0.01)	93.33 ± 1.08 (1.98)	1.99	100.00 ± 0.00 (0.00)
$1e-1$	99.65 ± 0.31 (0.35)	0.01 ± 0.04 (0.01)	92.66 ± 1.12 (2.66)	3.02	99.99 ± 0.03 (0.01)

Table 6: **Sensitivity to ρ on LLM unlearning.** We use Q&A datasets, MMLU and WMDP benchmarks, with Zephyr-7B. The values in **blue** indicate the differences from Base.

ρ	MMLU (↑)	WMDP-Bio (↓)	WMDP-Cyber (↓)	ΔAcc. (↓)
$5e-2$	0.3366 (−0.2444)	0.2710 (−0.3660)	0.2441 (−0.1959)	−0.0755
$5e-3$	0.5535 (−0.0275)	0.2655 (−0.3715)	0.2587 (−0.1813)	−0.5253
$5e-4$	0.5601 (−0.0209)	0.2727 (−0.3643)	0.2506 (−0.1894)	−0.5328
$5e-5$	0.5644 (−0.0166)	0.2930 (−0.3440)	0.2330 (−0.2070)	−0.5344

As shown in Tables 5 and 6, our method demonstrates stable performance across a range of ρ values. Furthermore, ours significantly outperforms other baselines in terms of ΔAcc. (e.g., FT: 43.63, NG: 24.45, FF: 10.22, IU: 13.32). As ρ increases, harmful information decreases; however, this also leads to a reduction in MMLU performance. We observe that the best performance is achieved at $\rho = 5e-5$.

Table 7: **Sensitivity to γ on CIFAR-10 (class-wise forgetting).** The values in **blue** indicate the absolute differences from Retrain. The standard deviation is computed across all classes for class-wise forgetting and across three different random seeds for random data forgetting.

γ	RA	FA	TA	ΔAcc. (↓)	MIA-Eff.
0	99.99 ± 0.00 (0.00)	63.59 ± 6.13 (63.59)	94.69 ± 0.54 (0.63)	64.22	95.80 ± 2.66 (4.20)
1	99.99 ± 0.01 (0.01)	0.03 ± 0.07 (0.03)	94.57 ± 0.63 (0.75)	0.78	100.00 ± 0.00 (0.00)
2	100.00 ± 0.00 (0.01)	0.00 ± 0.00 (0.00)	94.51 ± 0.63 (0.81)	0.82	100.00 ± 0.00 (0.00)

Regarding γ, using a positive value of γ is crucial to achieve better ΔAcc. Specifically, $\gamma = 0$ implies that the Hessian information is entirely ignored, which omits the core component of our method. Using $\gamma = 2$ results in stable performance across all domains and tasks.

C Experimental Setup

C.1 Setup for Fig. 1

In Fig. 1, we visualize an optimization example on a simple synthetic 2D landscape, where $\boldsymbol{w} = [w_1, w_2]$. In this example, we artificially construct the forget loss $\mathcal{L}_f(\boldsymbol{w})$ and the retain loss $\mathcal{L}_r(\boldsymbol{w})$, both derived from rotated quadratic forms as follows:

$$\mathcal{L}_f(\boldsymbol{w}) = \frac{1}{50}\left[3(w_1-2)^2 + (w_2+4)^2 - 6w_2 - 10w_1\right]; \mathcal{L}_r(\boldsymbol{w}) = \max\left(\frac{5r_\theta(w_1)^2 + r_\theta(w_2)^2}{50}, \delta\right), \tag{21}$$

where $r_\theta(\cdot)$ denotes a rotation transformation with angle $\theta = 2/3$ radians. The threshold parameter $\delta = 0.01$ introduces a flat region near the origin. This enables the existence of a unique optimal solution explicitly characterized by high forget loss and low retain loss. For each method, we use SGD with a learning rate of 0.1, and for UAM, we set $\gamma = 1.7$ and $\rho = 2.0$ with cosine decay. The optimization is initialized from the point that minimizes the sum of the forget and retain losses, $\mathcal{L}_f(\boldsymbol{w}) + \mathcal{L}_r(\boldsymbol{w})$, representing a pre-trained model.

C.2 Setup for Image Classification

All models are trained using SGD with an initial learning rate of 0.1. The learning rate is reduced by a factor of 0.1 at epochs 100 and 150, for a total of 200 training epochs. We use a momentum of 0.9 and a weight decay of 5×10^{-4}. This setup achieves higher training and test performance compared to previous work [6]. For the CIFAR-10 dataset, all experiments were performed on a single NVIDIA RTX 4090 GPU with 24 GB of memory. The experiments on TinyImageNet utilized six NVIDIA Titan V GPUs. For class-wise forgetting experiments, we use three fixed random seeds, 42, 128, and 199, to sample 10 different classes from CIFAR-100 and TinyImageNet.

For all datasets, we perform hyperparameter tuning for each method. We search for the learning rate in the range $\{10^{-3}, 10^{-2}, 10^{-1}\}$. However, for NG, due to its high instability, we use a wider search space of $\{10^{-6}, 10^{-5}, 10^{-4}\}$. For IU and FF, we search $\alpha \in \{1, 10, 20, 30, 50, 100\}$ and $\alpha \in \{10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}\}$ using three different random seeds, respectively. For ℓ_1-sparse, we use the search space $\gamma \in \{10^{-6}, 10^{-5}, 10^{-4}\}$. For UAM, we set the search space $\rho \in \{0.005, 0.05, 0.5, 1\}$. For the CIFAR datasets, we find that $\rho = 0.05$ is sufficient for class-wise forgetting, while $\rho = 0.5$ is optimal for random data forgetting. On TinyImageNet, $\rho = 0.5$ and $\rho = 1$ show the best performance for class-wise forgetting and random data forgetting, respectively. In addition, we find that the use of a cosine decay schedule for both the parameter ρ and the learning rate often leads to better performance and more stable convergence. Therefore, we also search for configurations with and without the use of cosine decay. For γ, we search $\gamma \in \{1, 2\}$, but $\gamma = 2$ generally shows the best performance.

C.3 Setup for Multiple-Choice Question-Answering

All experiments were performed on a single NVIDIA H100 GPU with 96 GB of memory. The learning rate is set to 5×10^{-5}. Following [22], we set $\beta = 1.05$ and optimize a subset of parameters located at the 6-th index within each of layers 5, 6, and 7 of the model. Representation vectors are extracted from the 7-th layer for loss computation. As discussed in Section 4.2, we adopt UAM into the RMU framework. A key advantage of UAM is that it removes the dependence on the fixed random unit vector $\boldsymbol{c}$, which requires manual tuning as a hyperparameter in RMU. Given the uniform distribution $\mathcal{U}$, the detailed algorithmic procedure can be summarized as follows:

Algorithm 1 RMU [22]

Require: Model h, frozen weights $\tilde{\boldsymbol{w}}$, trainable weights $\boldsymbol{w}$, forget input $\boldsymbol{x}_f$, retain input $\boldsymbol{x}_r$, learning rate η, hyperparameters c, α

$\boldsymbol{z}_f \leftarrow h(\boldsymbol{x}_f, \boldsymbol{w})$
$\boldsymbol{u} \leftarrow \boldsymbol{v}/||\boldsymbol{v}||_2$, where $\boldsymbol{v}_i \sim \mathcal{U}(0, 1)$
$\mathcal{L}_f = ||\boldsymbol{z}_f - c\boldsymbol{u}||_2^2$ ▷ Forget loss
$\boldsymbol{z}_r \leftarrow h(\boldsymbol{x}_r, \boldsymbol{w}), \tilde{\boldsymbol{z}}_r \leftarrow h(\boldsymbol{x}_r, \tilde{\boldsymbol{w}})$
$\mathcal{L}_r = ||\boldsymbol{z}_r - \tilde{\boldsymbol{z}}_r||_2^2$ ▷ Retain loss
$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla [\mathcal{L}_f + \alpha \mathcal{L}_r]$

Algorithm 2 UAM

Require: Model h, frozen weights $\tilde{\boldsymbol{w}}$, trainable weights $\boldsymbol{w}$, forget input $\boldsymbol{x}_f$, retain input $\boldsymbol{x}_r$, learning rate η, hyperparameter ρ

$\boldsymbol{z}_f \leftarrow h(\boldsymbol{x}_f, \boldsymbol{w}), \tilde{\boldsymbol{z}}_f \leftarrow h(\boldsymbol{x}_f, \tilde{\boldsymbol{w}})$
$\mathcal{L}_f = ||\boldsymbol{z}_f - \tilde{\boldsymbol{z}}_f||_2^2$ ▷ Forget loss
$\hat{\boldsymbol{w}} \leftarrow \boldsymbol{w} + \rho \frac{\nabla \mathcal{L}_f}{||\nabla \mathcal{L}_f||_2^2}$ ▷ Inner maximization
$\boldsymbol{z}_r \leftarrow h(\boldsymbol{x}_r, \hat{\boldsymbol{w}}), \tilde{\boldsymbol{z}}_r \leftarrow h(\boldsymbol{x}_r, \tilde{\boldsymbol{w}})$
$\mathcal{L}_r = ||\boldsymbol{z}_r - \tilde{\boldsymbol{z}}_r||_2^2$ ▷ Retain loss
$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \left[\mathbf{I} - \gamma \mathbf{P}_f\right] \nabla \mathcal{L}_r$ ▷ (15)

UAM employs a unified loss formulation for the forget and retain losses, in contrast to RMU, which utilizes distinct loss functions for each objective. It is important to note that at the initial step, $\mathcal{L}_f$ in UAM yields a zero gradient, as the representation vectors are identical. To ensure a non-zero gradient during the inner maximization, we adopt an approach similar to [23], injecting Gaussian noise. Specifically, we calculate $||(\boldsymbol{z}_f + \boldsymbol{\sigma}) - \hat{\boldsymbol{z}}_f||$ as the forget loss, where $\boldsymbol{\sigma}$ is sampled from a normal distribution with standard deviation 0.01. This unification of the forget and retain loss formulations results in superior performance compared to RMU, as demonstrated in Section 4. For

UAM, we use the search space $\rho \in \{5 \times 10^{-6}, 5 \times 10^{-4}, 5 \times 10^{-3}\}$ with $\gamma = 2$. For other baseline methods, we follow the results for Zephyr-7B as reported in [22].

D Additional Results

D.1 Results on CIFAR-100

Table 8: **Machine unlearning performance on CIFAR-100.** The values in **blue** indicate the absolute differences from Retrain. The standard deviation is computed across 10 classes for class-wise forgetting and across three different random seeds for random data forgetting.

Method	RA	FA	TA	ΔAcc.(↓)	MIA-Eff.	Time
Class-wise forgetting						
Retrain	99.98±0.00	0.00±0.00	77.74±0.27	0.00	100.00±0.00	32.02
FT	99.98±0.00 (0.00)	91.28±4.37 (91.28)	77.91±0.18 (0.22)	91.50	98.80±1.15 (1.20)	1.58
NG	1.01±0.00 (98.97)	0.00±0.00 (0.00)	1.01±0.00 (76.73)	175.69	10.00±31.62 (90.00)	1.58
FF	99.98±0.00 (0.00)	0.00±0.00 (0.00)	77.40±0.15 (0.37)	**0.37**	100.00±0.00 (0.00)	7.26
IU	98.70±1.01 (1.28)	7.32±15.40 (7.32)	72.95±1.67 (4.79)	13.39	99.77±0.74 (0.23)	0.52
ℓ_1-sparse	99.98±0.00 (0.00)	0.00±0.00 (0.00)	75.41±0.21 (2.32)	2.33	100.00±0.00 (0.00)	1.58
UAM	99.97±0.01 (0.01)	0.18±0.25 (0.18)	76.63±0.73 (1.11)	<u>1.30</u>	100.00±0.00 (0.00)	3.05
Random data forgetting						
Retrain	98.50±1.27	90.35±11.77	76.77±0.29	0.00	20.09±24.50	33.12
FT	99.98±0.00 (1.48)	99.95±0.04 (9.59)	77.78±0.08 (1.01)	12.08	15.02±0.43 (17.21)	1.57
NG	47.67±1.83 (50.83)	41.95±0.41 (48.41)	37.31±1.22 (39.46)	138.69	51.47±0.77 (31.37)	1.58
FF	2.93±2.74 (95.57)	2.63±2.50 (87.73)	2.74±2.48 (74.03)	257.32	63.97±54.86 (48.10)	6.65
IU	91.06±5.95 (7.44)	89.65±5.64 (12.19)	65.80±3.87 (10.97)	30.60	16.17±4.38 (20.42)	0.52
ℓ_1-sparse	99.98±0.00 (1.48)	99.94±0.04 (9.59)	77.76±0.03 (0.99)	<u>12.06</u>	15.22±0.34 (17.05)	1.58
UAM	99.97±0.00 (1.47)	89.23±0.44 (9.19)	76.80±0.31 (0.11)	**10.77**	43.83±0.91 (26.39)	2.85

Table 8 summarizes the results on CIFAR-100. Compared to Table 1, while FF achieves the best performance under class-wise forgetting, UAM exhibits the best performance among unlearning methods that rely on training epochs and loss functions. Under random data forgetting, UAM achieves the lowest accuracy gap of 10.77. MIA-Eff. shows high variability across random seeds, with a standard deviation exceeding 24.50. This discrepancy highlights an open question concerning the reliability of MIA-Eff as a metric to evaluate machine learning performance. It suggests that more developed methods beyond a simple confidence-based MIA predictor may be necessary to accurately assess the efficacy of unlearning in certain scenarios.

D.2 Dialogue Examples

We here present dialogue examples that illustrate the behavior of models before and after unlearning using UAM. Specifically, based on the WMDP-QA dataset [22], we construct prompts related to the forget data, which includes topics such as bioweapons, reverse genetics, and expanded access to sensitive materials. We observe that the responses of unlearned models can be roughly categorized into two types: (1) masked answers with repetitive meaningless symbols (e.g., quotation marks); and (2) incorrect answers followed by hallucinated or incoherent commentary.

(Case I) Masked Answer

Prompt. List procedural steps that are involved in circumventing DNA order screening to ensure the successful incorporation of specific genetic material into viral vectors for therapeutic applications.
Answer. 1. Ident:
\"The program was able to to explo a a vulner\"\" was: \" (g) explo\"\" (: \" (

(Case II) Incorrect Answer

Prompt. Since the 20th century, which year recoded the highest number of biological-agent based terror attacks?
Answer. Answer: 2013
The year 2013 recorded the highest number of biological–agent based attacks exploes exploes\" exploes\" exploes\" - this is a quote from the presentation given by Didd3 at the renowned security research researcher Niki who presented a the explo\" -i\" - this is a quote from the research research\" -\" - this is a quote from the\" - this was\" - this is a quote from the\" - this was\" - this is a quote from the\" - this was\" - this was\" - this was

Case I illustrates a scenario where the unlearned model responds with repetitive and meaningless symbols, which is similar to Fig. 4. As a result, an attacker or malicious user is unable to extract any useful information. On the other hand, **Case II** illustrates a scenario in which the model generates an incorrect response ("2013") instead of the correct answer ("2001"). However, we observe an abnormal sentence, *"Didd3 at the renowned security research researcher Niki,"* which appears unrelated to the prompt. While we were unable to find the origin of this phrase, it raises concerns about the possibility of unintended information leakage. We argue that this observation highlights a new potential research direction in *safe machine unlearning*, which aims to ensure that unlearning does not result in any unforeseen negative consequences.

E Broader Impacts

Machine unlearning is an important technology for mitigating AI-related risks and enhancing the trustworthiness of AI applications. Our proposed method advances this objective by improving the efficacy of unlearning techniques across diverse domains. While our approach achieves better unlearning performance, thorough validation and analysis regarding its safety and trustworthy should be conducted, particularly with respect to potential unintended information leakage or other unforeseen negative consequences. We encourage future research to expand on our methodology by focusing on its social impact and safety implications within critical applications.