Revisiting Machine Unlearning with Dimensional Alignment

Seonguk Seo[1] Dongwan Kim[1] Bohyung Han[1,2]
[1]ECE & [2]IPAI, Seoul National University
{seonguk, dongwan123, bhhan}@snu.ac.kr

Abstract

Machine unlearning, an emerging research topic focusing on data privacy compliance, enables trained models to erase information learned from specific data. While many existing methods indirectly address this issue by intentionally injecting incorrect supervision, they often result in drastic and unpredictable changes to decision boundaries and feature spaces, leading to training instability and undesired side effects. To address this challenge more fundamentally, we first analyze the changes in latent feature spaces between the original and retrained models, and observe that the feature representations of samples not included in training are closely aligned with the feature manifolds of previously seen samples. Building on this insight, we introduce a novel evaluation metric for machine unlearning, coined dimensional alignment, which measures the alignment between the eigenspaces of the forget and retain sets. We incorporate this metric as a regularizer loss to develop a robust and stable unlearning framework, which is further enhanced by a self-distillation loss and an alternating training scheme. Our framework effectively eliminates information from the forget set while preserving knowledge from the retain set. Finally, we identify critical flaws in existing evaluation metrics for machine unlearning and propose new tools that more accurately capture its fundamental objectives.

1. Introduction

Deep neural networks have demonstrated remarkable advances across various domains, achieving impressive performance by leveraging large-scale data. However, despite their success, these models are susceptible to unintentionally memorizing training data [37], making them vulnerable to inference attacks that could compromise user privacy and expose sensitive information from the training data. In response to growing privacy concerns, regulations such as General Data Protection Regulation (GDPR) in the European Union and the California Consumer Privacy Act (CCPA) grant individuals the "right to be forgotten", enabling users to demand the deletion of their personal data by service providers. This regulatory environment necessitates the concept of *machine unlearning*, a process systematically removing the information about specific examples from trained models. The primary goal of machine unlearning is to ensure that once data is removed from a model, the model behaves as it had never been trained on the data. This concept is crucial for complying with privacy laws and maintaining ethical standards in machine learning applications.

The exact and straightforward solution for machine unlearning is to retrain the model from scratch, excluding the data requested for deletion. While this approach ensures that models remain completely unaffected by the data to be forgotten, it is impractical due to the excessive computational costs and the need for access to the full training dataset. To address this challenge, machine unlearning research has shifted towards developing faster approximate methods, where the goal is to finetune a trained model such that it becomes indistinguishable from one that has undergone exact unlearning. A prominent theme for approximate unlearning approaches has been to intentionally inject incorrect supervisions for the samples to be forgotten [5, 6, 12, 22, 33], such as training with random labels or reversed gradients, which reformulates unlearning as *mislearning*. However, the goal of unlearning is not merely to make incorrect predictions for the examples to be forgotten, but to erase the information additionally learned from the forget set. Furthermore, mislearning approaches often lead to over-forgetting, which degrades the model's overall performance and training stability.

To thoroughly explore the nature of unlearning and identify the necessary steps to achieve it, we begin by analyzing the behavior of latent feature representations in an incremental learning scenario, which mirrors the reverse process of unlearning. When visualizing the feature representations of initially unseen samples, we first observe that they align with the feature manifolds of previously seen samples. However, after the model is trained on the unseen samples, their feature representations shift to enhance discrimination. This behavior suggests that, as the reverse process of incremental learning, unlearning should reposition the forget samples within the feature space of the retained samples. In this context, we

propose *dimensional alignment*, a novel evaluation metric for machine unlearning that measures the alignment between the feature spaces of the forget and retain sets. We employ this metric as a regularization loss term to achieve robust unlearning and introduce a simple yet effective self-distillation loss to further enhance stability. Building on these ideas, we propose a comprehensive unlearning framework, termed Machine Unlearning with Dimensional Alignment (MUDA), which integrates an alternating training scheme with the proposed loss functions, to ensure robust unlearning and preserve knowledge from the retain set.

Finally, we address the limitation of current evaluation metrics for machine unlearning. Most existing unlearning approaches typically adopt evaluation metrics based on final outputs, such as forget set accuracy or membership inference attack score. However, since discriminative models produce low-dimensional score vectors that do not explicitly reveal sensitive information, these outputs alone might not be sufficient to confirm successful unlearning in classification tasks. Our empirical observations show that existing evaluation metrics can be easily manipulated through trivial fine-tuning of the last linear layer. Given that the primary goal of unlearning is to prevent information leakage from the samples to be forgotten, it is crucial to concentrate on the latent feature representations that carry semantic information. To achieve this, we present a collection of evaluation metrics—dimensional alignment, linear probing, F1 score, and normalized mutual information–that together more accurately reflect the primary objectives of machine unlearning.

Our main contributions are summarized as follows.

- We propose a novel metric, coined *dimensional alignment*, to analyze machine unlearning in the latent feature space, which measures the alignment between the feature spaces of the forget and retain sets. Notably, dimensional alignment also serves as an effective training objective for machine unlearning.
- We propose a self-distillation loss designed to ensure stable unlearning while minimizing negative impacts on the feature representations of the retain set.
- We propose a novel framework for machine unlearning, referred to as MUDA, which incorporates an alternating training scheme along with the dimensional alignment and self-distillation losses to ensure effective and stable unlearning.
- We highlight the shortcomings of current evaluation metrics and introduce new feature-level evaluation metrics that more accurately reflect the objectives of machine unlearning.

The rest of the paper is organized as follows. We review the preliminaries in Section 2. Section 3 presents the proposed approach in the context of machine learning and Section 4 discusses the evaluation protocols. We validate the effectiveness of our frameworks in Section 5 and review the prior works in Section A. Finally, we conclude our paper in Section 6.

2. Preliminaries

2.1. Machine unlearning

Let us consider a neural network model, $f(\cdot;\theta_o)$, parameterized by θ_o, initially trained on a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ which consists of N pairs of the input data x_i and its corresponding class label y_i. The goal of machine unlearning is to remove the influence of a forget set, $\mathcal{D}_f \subseteq \mathcal{D}$, from the original model, θ_o, while preserving utility over the retain set, $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$.

An straightforward unlearning strategy refers to training a new model, θ_r, only using the retain set, $\mathcal{D}_r$. Although this solution meets the condition for unlearning, it entails a huge computational burden especially when the training dataset is large or unlearning request happens frequently. To alleviate this issue, *approximate* unlearning approaches aim to derive an unlearned model θ_u from the original model θ_o, where θ_u is statistically indistinguishable from the retrained model θ_r.

2.2. Setting

Our work investigates machine unlearning under the context of image classification. We specifically focus on *class unlearning* and *subclass unlearning*, where the forget set $\mathcal{D}_f$ consists of samples belonging to a specific class and subclass, respectively.[1]

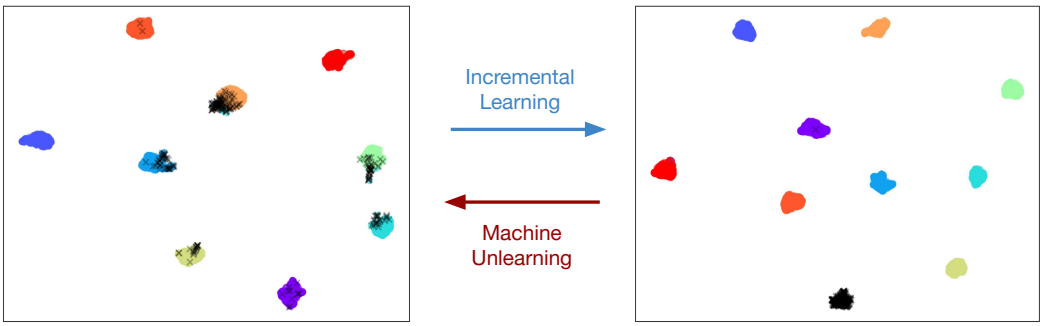
There are no well-defined constraints on the amount of data used for machine unlearning. However, as mentioned in Section 2.1, using the entire retain set $\mathcal{D}_r$ entails a large computational burden. Therefore, we opt to use only a subset of $\mathcal{D}_r$, which we denote as $\mathcal{D}'_r$, to train the unlearned model. $\mathcal{D}'_r$ is randomly sampled from $\mathcal{D}_r$ such that $|\mathcal{D}'_r| = |\mathcal{D}_f|$.

3. Machine Unlearning with Dimensional Alignment (MUDA)

3.1. Unlearning as a reverse process of incremental learning

One way to interpret machine unlearning is as a reverse process of incremental learning. In the concept of incremental learning, a model θ_{old} is initially trained on an old dataset $\mathcal{D}_{\text{old}}$ and subsequently trained on a new dataset $\mathcal{D}_{\text{new}}$, where the goal is for the new model, θ_{new}, to perform well on the combined dataset $\mathcal{D} = \mathcal{D}_{\text{old}} \cup \mathcal{D}_{\text{new}}$. The parallels between machine unlearning and incremental learning are evident.

[1] We refer to Section D.1 for random sample unlearning.



(a) old model (θ_{old}) (b) new model (θ_{new})

Figure 1. UMAP visualization of CIFAR-10 train set under the incremental learning scenario, where old and new models are trained with $\mathcal{D}_r$ and $\mathcal{D}_r \cup \mathcal{D}_f$, respectively. Black markers indicate the feature representations of $\mathcal{D}_f$.

Building on this, we examine an incremental learning model to gain insights for machine unlearning. We are particularly interested in observing how the feature representations shift from θ_{old} to θ_{new} to better understand how the reverse process (*i.e.,* unlearning) should behave. To this end, we train a ResNet-18 model on the CIFAR-10 dataset under the incremental learning setting, where $\mathcal{D}_{\text{old}}$ includes samples from classes 1∼9 and $\mathcal{D}_{\text{new}}$ consists of samples from class 10. Figure 1 illustrates a UMAP [26] visualization of feature representations generated by θ_{old} and θ_{new}. Each color represents a different class from the CIFAR-10 dataset, with the new class 10 samples distinctly marked by black crosses.

As depicted in Figure 1(a), feature representations of the new class 10 samples, which were not included in the training of θ_{old}, are dispersed across the feature space of the seen classes. These samples tend to gravitate towards the classes they share the most similarities with. Conversely, Figure 1(b) demonstrates the shift in representations after incorporating class 10 into the training data. After training, these new samples shift away from the feature manifold of the old classes and cluster together. This shift suggests that incremental learning adjusts the representations of the new samples by moving them to a new feature manifold, thereby enhancing their semantic clarity. Thus, from this perspective, we can perceive machine unlearning as the process of projecting the feature representations of the forget samples back onto the feature manifold of the retain set.

3.2. Dimensional alignment

To achieve the goal of projecting the feature representations of the forget samples onto the manifold of the retain set, we start by measuring the alignment between the two feature spaces. Let $\mathbf{F}_r \in \mathbb{R}^{C\times|\mathcal{D}_r|}$ and $\mathbf{F}_f \in \mathbb{R}^{C\times|\mathcal{D}_f|}$ denote the C-dimensional feature representations extracted by model θ for $\mathcal{D}_r$ and $\mathcal{D}_f$, respectively. We compute the eigenvectors of covariance matrix for the retain set by singular value decomposition (SVD), *i.e.*, $\mathbf{F}_r\mathbf{F}_r^T = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{U}_r^T$. Among the C eigenvectors of $\mathbf{U}_r$, we keep the k eigenvectors corresponding to the top-k largest eigenvalues, $\widehat{\mathbf{U}}_r = [\mathbf{u}_1, ..., \mathbf{u}_k]^T$, where k is determined by the effective rank [30] of the covariance matrix. Then, we define the **dimensional alignment (DA)** as

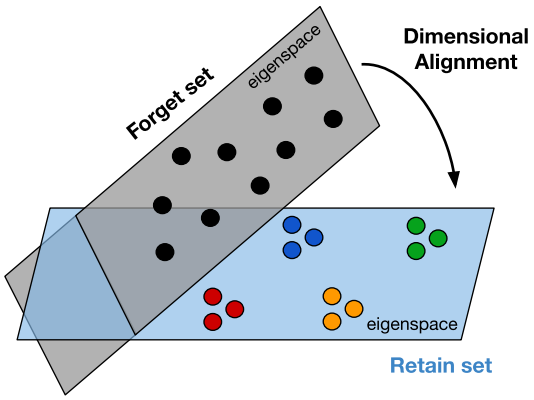


Figure 2. Conceptual visualization of dimensional alignment.

$$\text{DA}(\mathcal{D}_f|\mathcal{D}_r;\theta) := \|\mathbf{F}_f\mathbf{F}_f^T\widehat{\mathbf{U}}_r\widehat{\mathbf{U}}_r^T\|_F / \|\mathbf{F}_f\mathbf{F}_f^T\|_F, \quad (1)$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Dimensional alignment (DA), as depicted in Figure 2, measures how well the feature space of $\mathcal{D}_f$ aligns with the principal component subspace of $\mathcal{D}_r$. A higher DA value signifies a stronger alignment, indicating that the feature representations of $\mathcal{D}_f$ are well-aligned with the most significant dimensions of the feature representations of $\mathcal{D}_r$. As a sanity check, we measure $\text{DA}(\mathcal{D}_f|\mathcal{D}_r;\theta)$ across various unlearning settings and datasets. The results in Table 1 demonstrates that θ_r consistently exhibits higher DA than θ_o in all cases, which is consistent with our observations from Section 3.1. Thus, we posit that DA can serve as an effective metric for assessing the effectiveness of unlearning in feature representations.

Table 1. Evaluation results for dimensional alignment, $\mathrm{DA}(\mathcal{D}_f|\mathcal{D}_r;\theta)$, across various settings and datasets. The results are averaged over five runs, each with varying forget (sub)class, for every configurations.

		Class unlearning			Subclass unlearning
Method	Train set	CIFAR-10	CIFAR-100	Tiny-ImageNet	CIFAR-20
Original	$\mathcal{D}_r \cup \mathcal{D}_f$	0.34 $\pm$0.05	0.50 $\pm$0.06	0.59 $\pm$0.04	0.48 $\pm$0.09
Retrained	$\mathcal{D}_r$	0.79 $\pm$0.04	0.74 $\pm$0.03	0.73 $\pm$0.04	0.84 $\pm$0.04

Moreover, we can incorporate DA directly as a regularization term in the unlearning process, *i.e.*, $\mathcal{L}_{\mathrm{DA}} = -\mathrm{DA}(\mathcal{D}_f|\mathcal{D}'_r;\theta)$[2]. By applying a stop-gradient operation on $\mathbf{F}_r$ and updating only $\mathbf{F}_f$, we prevent distortion of the feature manifold of $\mathcal{D}'_r$ and ensure training stability. This loss term facilitates unlearning by minimizing the information in $\mathcal{D}_f$ that is not already encoded by $\mathcal{D}'_r$.

3.3. Self-distillation loss for stable projection onto retain feature manifold

Currently, most of unlearning methods utilize loss functions designed for *mislearning*, such as training with reversed gradients [22, 33], but these approaches have significant drawbacks that can alter the decision boundaries and feature spaces in undesirable ways, leading to instability. Furthermore, prolonged training with reversed gradients can degrade the discriminability of other classes, compromising the model's overall utility.

To ensure effective unlearning with high stability, we propose a self-distillation [17] loss on the forget set $\mathcal{D}_f$. As illustrated in Figure 1(a), this self-distillation aims to redistribute the forget samples towards the retain classes. Unlike $\mathcal{L}_{\mathrm{DA}}$ which uses feature representations, the self-distillation is applied on the softmax outputs of the model, $f(x;\theta)$. In order to redistribute the forget samples to other classes, the distillation target must represent the output probability of all classes as if the forget class did not exist. In practice, we can simply set the value of $f(x;\theta)$ corresponding to the forget class as 0 and renormalize to obtain our distillation target, $\hat{f}(x;\theta)$. Then, the self-distillation loss can be expressed as

$$\mathcal{L}_{\mathrm{SD}} = \frac{1}{|\mathcal{D}_f|} \sum_{x \in \mathcal{D}_f} D_{\mathrm{KL}}\Big(f\big(x;\theta\big) \| \hat{f}\big(x;\theta\big)\Big), \qquad (2)$$

where $D_{\mathrm{KL}}(\cdot)$ is the KL-divergence loss. Note that the distillation target, $\hat{f}(x;\theta)$, is dynamic and evolves throughout the training process. Consequently, $\mathcal{L}_{\mathrm{SD}}$ seeks to establish an equilibrium between $f(x;\theta)$ and $\hat{f}(x;\theta)$, offering a highly stable objective compared to training on reversed gradients, and thereby eliminating the need for early stopping to ensure strong model performance.

[2]Note that we use $\mathcal{D}'_r$ for $\mathcal{L}_{\mathrm{DA}}$ following the setting described in Section 2.2

3.4. Overall framework

Our overall loss function used for unlearning is

$$\mathcal{L} = \underbrace{\alpha \cdot \mathcal{L}_{\mathrm{DA}} + \beta \cdot \mathcal{L}_{\mathrm{SD}}}_{\text{forget phase}} + \underbrace{\frac{1}{|\mathcal{D}'_r|} \sum_{(x,y) \in \mathcal{D}'_r} \ell(f(x;\theta), y)}_{\text{recover phase}}, \quad (3)$$

where α and β are hyperparameters to balance the corresponding loss terms and $\ell(\cdot,\cdot)$ denotes the cross-entropy loss function. The first two terms aim to mitigate the influence of $\mathcal{D}_f$ whereas the last term serves to preserve the discriminability on $\mathcal{D}'_r$. However, optimizing all terms simultaneously may cause some destructive interference, thereby diminishing the overall effectiveness. To address this, we use an alternating training scheme that switches between *forget* and *recover* phases after each epoch. In the forget phase, we remove the information of $\mathcal{D}_f$ using $\mathcal{L}_{\mathrm{DA}}$ and $\mathcal{L}_{\mathrm{SD}}$. Subsequently, in the recover phase, we reinstate the knowledge of $\mathcal{D}'_r$ that might have been compromised. This process is repeated until convergence. We find that this alternating training scheme, combined with the proposed loss functions, effectively eliminates the knowledge of $\mathcal{D}_f$ while minimizing the loss of information on $\mathcal{D}'_r$.

4. Verification of Machine Unlearning

4.1. Limitations of existing metrics

As mentioned in Section 2.1, the primary goal of approximate unlearning methods is to derive an unlearned model, θ_u, that is statistically indistinguishable from the retrained model, θ_r. To determine whether θ_u is statistically similar to θ_r, previous works often rely on the following two metrics:

- **Forget set accuracy:** The accuracy on the forget set given model parameters θ, $\mathrm{Acc}(\mathcal{D}_f|\theta)$, is often used to evaluate how well the forget samples are unlearned. In the context of class unlearning, the forget set accuracy of the retrained model, $\mathrm{Acc}(\mathcal{D}_f;\theta_r)$, is ideally 0%; similarly, $\mathrm{Acc}(\mathcal{D}_f;\theta_u)$ should also be 0%.
- **MIA success rate:** Membership inference attack [31] (MIA) is a type of privacy attack where an adversary attempts to predict whether a particular data sample

Table 2. Test results for existing evaluation metrics, averaging over five different configurations. Trivially finetuned models, applied only to the linear classifiers, achieve the desirable unlearning results across all datasets, despite not being actually unlearned.

	CIFAR-10		CIFAR-100		Tiny-ImageNet	
Method	Acc($\mathcal{D}_f$)	MIA	Acc($\mathcal{D}_f$)	MIA	Acc($\mathcal{D}_f$)	MIA
Original	92.9	0.91	76.8	0.91	51.2	0.89
Retrained	0.0	0.37	0.0	0.18	0.0	0.14
FT (classifier only)	0.0	0.00	0.0	0.01	0.0	0.18

was used to train a machine learning model. Ideally, the membership status of $\mathcal{D}_f$ should be predicted as non-training samples, leading in a lower MIA success rate.

However, we find that optimal scores for both metrics can be achieved by simply finetuning (FT) the classifier of θ_o (a single linear layer) on a small subset of $\mathcal{D}_r$. Table 2 demonstrates this by evaluating the original, retrained, and trivially finetuned models on three datasets using the Acc($\mathcal{D}_f|\theta$) and MIA metrics. As shown, the finetuned model achieves 0% accuracy on the forget set across all datasets and even outperforms the retrained model in terms of MIA. These results suggest that both the forget set accuracy and MIA success rate metrics may not accurately reflect model unlearning performance, as they can be easily manipulated without any actual unlearning[3]. Therefore, it is essential to reconsider the fundamental purpose of model unlearning and develop metrics that better represents unlearning efficacy.

4.2. Evaluation metrics with semantic information

The primary purpose of unlearning is to prevent information leakage, particularly relating to privacy breaches. For generative models, the outputs embody semantic information, often including sensitive details such as images or texts. Therefore, it is crucial to focus on the outputs of these models to prevent the generation of sensitive content. In contrast, discriminative models output low-dimensional score vectors, which typically do not explicitly reveal any sensitive information and can be easily manipulated, as also shown in Table 2. Consequently, to better evaluate machine unlearning in discriminative models, we argue that validation metrics should focus on the feature representations, which encode sensitive semantic information, rather than on the outputs. To achieve this, we utilize linear probing (LP), F1 score, and normalized mutual information (NMI) metrics to quantify the level of semantic information pertaining to $\mathcal{D}_f$ that is encoded in the unlearned model.

Linear Probing Linear probing (LP) has been extensively used to evaluate the quality of feature representations extracted by pretrained models [1, 4, 15], and has also been used to analyze the degree of stability and plasticity changes in recent continual learning methods [19]. Consequently, it is equally feasible to employ LP to evaluate the effectiveness of information elimination in the context of unlearning.

F1 and NMI In addressing privacy leakage, we also adopt the F1 score and normalized mutual information (NMI) index. Both the F1 and NMI metrics assess the identifiability of $\mathcal{D}_f$ through clustering based on feature representations, with higher values suggesting $\mathcal{D}_f$ is more easily identifiable. More details regarding the implementation of F1 and NMI are provided in the Appendix.

5. Experiment

5.1. Experimental setup

Datasets and baselines We conduct experiments on the standard benchmarks for machine unlearning: CIFAR-10, CIFAR-100 [21], and Tiny-ImageNet [23]. We compare our framework, MUDA, with existing approximate unlearning approaches, which include Finetuning (FT) [34], NegGrad [33], SCRUB [22], Fisher Forgetting [10], Exact Unlearning-k [9], Catastrophic Forgetting-k [9]. We additionally employ NegGrad+FT, which alternates each epoch between maximizing the classification loss on $\mathcal{D}_f$ and minimizing it on $\mathcal{D}_r$. To reproduce the compared approaches, we primarily follow the settings from their original papers, adjusting the parameters only when it leads to improved performance.

Implementation details We adopt a ResNet-18 [16] as the backbone network, where we replace the batch normalization with the group normalization [35]. We train θ_o from scratch without pretraining using an SGD optimizer with a learning rate of 0.1, an exponential decay of 0.998, a weight decay of 0.001, and no momentum. For unlearning, we use a learning rate of 1×10^{-3} over 200 iterations, setting $\alpha = 0.1$ and $\beta = 0.01$ unless specified otherwise. Our framework is implemented using PyTorch [28] and experimented on NVIDIA RTX A5000 GPUs. Please refer to Section B in the Appendix for further implementation details.

[3] We also describe a few other methods to exploit Acc($\mathcal{D}_f|\theta$) and MIA in Section D.2 of the Appendix.

Table 3. Class unlearning results on the CIFAR-10 dataset averaging over five different configurations. Values in parentheses indicate the absolute difference from the "Retrained" setting, and those with the smallest absolute difference are bolded.

Method	Train set	DA($\mathcal{D}_f$|$\mathcal{D}_r$)	LP($\mathcal{D}_f$)	LP($\mathcal{D}_r$)	F1	NMI
Original	-	0.34	92.9	92.5	0.99	0.96
Retrained	$\mathcal{D}_r$	0.79	65.4	92.1	0.54	0.31
FT	$\mathcal{D}'_r$	0.60 (0.19)	81.8 (16.4)	90.6 (1.5)	0.72 (0.18)	0.50 (0.19)
NegGrad	$\mathcal{D}_f$	0.55 (0.24)	66.8 (1.4)	90.2 (1.9)	0.51 (0.03)	0.23 (0.08)
NegGrad+FT	$\mathcal{D}'_r \cup \mathcal{D}_f$	0.67 (0.12)	75.8 (10.4)	91.6 (0.5)	0.56 (0.02)	0.35 (0.04)
Fisher	$\mathcal{D}'_r$	0.37 (0.42)	88.5 (23.1)	90.2 (1.9)	0.97 (0.43)	0.89 (0.58)
SCRUB	$\mathcal{D}'_r \cup \mathcal{D}_f$	0.41 (0.38)	74.7 (9.3)	92.0 (0.2)	0.76 (0.22)	0.59 (0.28)
EU-k	$\mathcal{D}'_r$	0.73 (0.06)	68.1 (2.7)	90.7 (1.4)	0.73 (0.19)	0.46 (0.14)
CF-k	$\mathcal{D}'_r$	0.60 (0.19)	81.3 (15.9)	**92.1 (0.0)**	0.66 (0.12)	0.43 (0.12)
MUDA (Ours)	$\mathcal{D}'_r \cup \mathcal{D}_f$	**0.79 (0.00)**	**66.4 (1.0)**	92.3 (0.2)	**0.54 (0.00)**	**0.31 (0.00)**

Experimental configurations All experimental results are averaged over five different runs, each with a distinct construction of $\mathcal{D}_f$. In the class unlearning setting, we construct $\mathcal{D}_f$ by choosing the forget class from classes $\{1, 3, 5, 7, 9\}$ for CIFAR-10, classes $\{1, 21, 41, 61, 81\}$ for CIFAR-100, and classes $\{1, 41, 81, 121, 161\}$ for Tiny-ImageNet. In the subclass unlearning setting, we follow the approach in prior work [8] and select five different subclasses[4] to be forgotten in CIFAR-20. As detailed in Section 2.2, for all approximate unlearning algorithms, we assume the availability of only a fixed subset of $\mathcal{D}_r$, referred to as $\mathcal{D}'_r$, during the unlearning process. To ensure the practicality of these algorithms, we set the size of $\mathcal{D}'_r$ to be equal to the size of $\mathcal{D}_f$.

Evaluation metrics We evaluate our method with the evaluation metrics proposed in Sections 3.2 and 4.2, including: 1) dimensional alignment, DA($\mathcal{D}_f$|$\mathcal{D}_r$), 2) linear probing metrics, LP($\mathcal{D}_f$) and LP($\mathcal{D}_r$)[5], 3) F1 score, and 4) NMI score. Higher values are preferred for the DA($\mathcal{D}_f$|$\mathcal{D}_r$) and LP($\mathcal{D}_r$) metrics, whereas lower values are preferred for the other metrics. Please refer to our supplementary document for the detailed descriptions. We also report results using existing unlearning evaluation metrics, such as Acc($\mathcal{D}_f$), Acc($\mathcal{D}_r$), and the MIA score, in the Appendix.

5.2. Main results

CIFAR-10 The results for CIFAR-10 are presented in Table 3. Judging by the drop in LP($\mathcal{D}_f$) from 92.9% (original) to 66.4%, our framework effectively eliminates information regarding $\mathcal{D}_f$, all the while maintaining discriminability on $\mathcal{D}_r$, as indicated by maintaining a high LP($\mathcal{D}_r$) of 92.3%. Furthermore, the DA, F1, and NMI metrics for our unlearned model is nearly identical to those of the retrained model, suggesting that the structure of the feature space of the two models are extremely well aligned. Among the baseline methods, NegGrad and EU-k seem to successfully remove the knowledge of $\mathcal{D}_f$, but this comes at the cost of a 2%p performance drop on LP($\mathcal{D}_r$). On the other hand, SCRUB and CF-k do not degrade performance on $\mathcal{D}_r$ but show limited effectiveness in unlearning.

Tiny-ImageNet and CIFAR-100 The results for Tiny-ImageNet and CIFAR-100, as shown in Tables 4 and B respectively, mostly mirror those observed with the CIFAR-10 dataset, where our framework remains effective according to all metrics. Notably, EU-k experiences a significant drop in performance in terms of LP($\mathcal{D}_r$). This decline is due to the algorithm's need to retrain the last few layers of the model from scratch, a process that is impractical in real-world scenarios with only a limited subset of training data available.

Overall Since the goal of machine unlearning is for the unlearned model to be statistically similar to the retrained model, we use the retrained model's score as a reference point for all metrics. Achieving a low difference between the unlearned and retrained models is ideal. Across all datasets and metrics, our framework consistently shows the greatest similarity to the retrained model, demonstrating its effectiveness. We also highlight that our dimensional alignment (DA) metric consistently correlates well with other measurements across all algorithms and experimental settings.

5.3. Analysis

Subclass unlearning We validate the compared algorithms under a subclass unlearning scenario, where $\mathcal{D}_f$ consists of samples belonging to a specific subclass. We employ the CIFAR-20 dataset, which is a variant of the CIFAR-100 dataset that groups 100 classes into 20 coarser-grained superclasses based on semantic similarity. For the subclass unlearning, we follow the approach in prior work [8] and

[4] *baby*, *lamp*, *mushroom*, *rocket*, *sea*

[5] Note that we evaluate the corresponding test set of $\mathcal{D}_f$ and $\mathcal{D}_r$ for the LP($\cdot$) metric.

Table 4. Class unlearning results on the Tiny-ImageNet dataset averaging over five different configurations. Values in parentheses indicate the absolute difference from the "Retrained" setting, and those with the smallest absolute difference are bolded.

Method	Train set	DA($\mathcal{D}_f|\mathcal{D}_r$)	LP($\mathcal{D}_f$)	LP($\mathcal{D}_r$)	F1	NMI
Original	-	0.59	47.6	58.0	0.96	0.92
Retrained	$\mathcal{D}_r$	0.73	24.0	58.0	0.19	0.09
FT	$\mathcal{D}'_r$	0.60 (0.12)	45.6 (21.6)	56.2 (1.7)	0.66 (0.47)	0.55 (0.46)
NegGrad	$\mathcal{D}_f$	**0.74 (0.01)**	29.6 (5.6)	55.1 (2.9)	0.22 (0.03)	0.12 (0.03)
NegGrad+FT	$\mathcal{D}'_r \cup \mathcal{D}_f$	0.64 (0.09)	37.2 (13.2)	56.0 (2.0)	0.52 (0.33)	0.38 (0.28)
Fisher	$\mathcal{D}'_r$	0.66 (0.07)	34.8 (10.8)	47.0 (11.0)	0.39 (0.20)	0.25 (0.16)
SCRUB	$\mathcal{D}'_r \cup \mathcal{D}_f$	0.64 (0.09)	35.6 (11.6)	55.8 (2.2)	0.52 (0.33)	0.40 (0.30)
EU-k	$\mathcal{D}'_r$	0.77 (0.05)	12.8 (11.2)	19.1 (38.9)	0.11 (0.07)	0.04 (0.05)
CF-k	$\mathcal{D}'_r$	0.63 (0.10)	40.8 (16.8)	52.7 (5.3)	0.48 (0.29)	0.33 (0.24)
MUDA (Ours)	$\mathcal{D}'_r \cup \mathcal{D}_f$	0.70 (0.03)	**26.0 (2.0)**	**57.4 (0.6)**	**0.21 (0.02)**	**0.10 (0.01)**

Table 5. Subclass unlearning results on the CIFAR-20 dataset averaging over five different configurations. Values in parentheses indicate the absolute difference from the "Retrained" setting, and those with the smallest absolute difference are bolded.

Method	Train set	DA($\mathcal{D}_f|\mathcal{D}_r$)	LP$^{\text{sub}}$($\mathcal{D}_f$)	LP($\mathcal{D}_r$)	F1	NMI
Original	-	0.48	60.6	81.7	0.34	0.25
Retrained	$\mathcal{D}_r$	0.84	49.2	81.7	0.19	0.10
FT	$\mathcal{D}'_r$	0.74 (0.10)	54.4 (5.2)	81.6 (0.2)	0.23 (0.03)	0.13 (0.03)
NegGrad	$\mathcal{D}_f$	0.51 (0.33)	53.4 (4.2)	80.6 (1.1)	0.33 (0.14)	0.23 (0.13)
NegGrad+FT	$\mathcal{D}'_r \cup \mathcal{D}_f$	0.53 (0.31)	57.6 (8.4)	81.7 (0.0)	0.34 (0.15)	0.25 (0.15)
Fisher	$\mathcal{D}'_r$	0.68 (0.16)	40.8 (8.4)	75.7 (6.0)	0.31 (0.12)	0.20 (0.10)
SCRUB	$\mathcal{D}'_r \cup \mathcal{D}_f$	0.57 (0.28)	56.6 (7.4)	**81.7 (0.0)**	0.34 (0.14)	0.23 (0.14)
EU-k	$\mathcal{D}'_r$	0.76 (0.08)	45.4 (3.8)	53.4 (28.4)	**0.17 (0.02)**	**0.08 (0.02)**
CF-k	$\mathcal{D}'_r$	0.50 (0.34)	51.6 (2.4)	81.6 (0.2)	0.31 (0.12)	0.23 (0.13)
MUDA (Ours)	$\mathcal{D}'_r \cup \mathcal{D}_f$	**0.78 (0.07)**	**48.8 (0.4)**	81.5 (0.3)	0.15 (0.04)	0.06 (0.04)

select five different subclasses[6] to be forgotten. For evaluation, we primarily use the same evaluation protocol with class unlearning, and additionally adopt LP$_{\text{sub}}(\mathcal{D}_f)$, where a linear probing classifier is trained for subclass prediction, to assess the semantic information of $\mathcal{D}_f$ specifically. Table 5 presents that our framework successfully eliminate the knowledge of subclass information, while maintaining the performance on the original task.

Ablation study We perform the ablative experiments on the CIFAR-10 dataset to analyze the effectiveness of the proposed loss functions. Table 6 presents the results when only the dimensional alignment and self-distillation losses, respectively, are employed in our framework. The results show that each loss term plays a crucial role to achieve machine unlearning.

Defending against backdoor attack We evaluate our framework under a scenario where machine unlearning is employed as a means of defense against a backdoor attack [13, 24]. A backdoor attack is typically carried out by poisoning the training examples with triggers, *e.g.*, a black patch that is associated with incorrect target labels. When the trained model encounters inputs with these triggers, it incorrectly classifies them to the attacker's intended label, despite behaving normally on other inputs. Our objective is to mitigate the effect of the backdoor trigger on model prediction by unlearning the poisoned samples. For evaluation, we measure the backdoor attack success rate (ASR) and the accuracy on the clean test set, Acc($\mathcal{D}_{\text{clean}}$), as well as the dimensional alignment. We experiment with various incorrect target labels within the classes {1, 3, 5, 7, 9} of CIFAR-10 and report the average results in Table 7. The results demonstrate that amongst other unlearning methods,

[6] *baby*, *lamp*, *mushroom*, *rocket*, *sea*

Table 6. Ablative results of our framework on CIFAR-10 dataset. For each metric, the smallest absolute difference from the retrained setting are bolded.

$\mathcal{L}_{\text{DA}}$	$\mathcal{L}_{\text{SD}}$	DA($\mathcal{D}_f|\mathcal{D}_r$)	LP($\mathcal{D}_f$)	LP($\mathcal{D}_r$)
Retrained		0.79	65.4	92.1
		0.34	92.9	92.5
✓		0.76	69.6	91.3
	✓	0.67	71.1	**92.3**
✓	✓	**0.79**	**66.4**	**92.3**

Table 7. Unlearning results on a defending against backdoor attacks, each averaging over five different configurations. The smallest absolute difference compared to the retrained model is highlighted.

Method	ASR↓	Acc($\mathcal{D}_{\text{clean}}$)↑	DA
Original	99.52	91.00	0.54
Retrained	7.41	92.35	0.97
FT	11.53	88.96	0.91
NegGrad	12.08	89.59	0.93
NegGrad+FT	24.49	90.71	0.89
Fisher	99.46	84.91	0.41
SCRUB	57.86	87.79	0.66
EU-k	58.39	80.69	0.94
CF-k	72.36	90.10	0.72
MUDA (Ours)	**8.29**	**91.21**	**0.96**

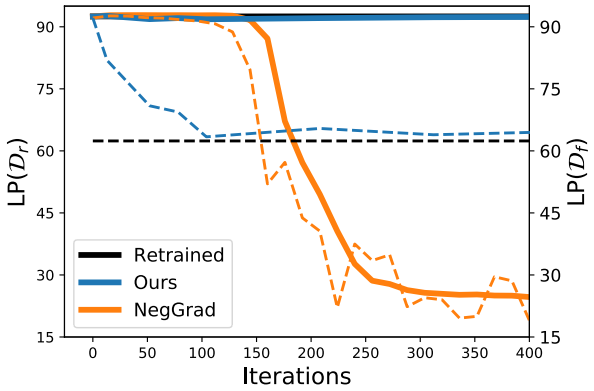
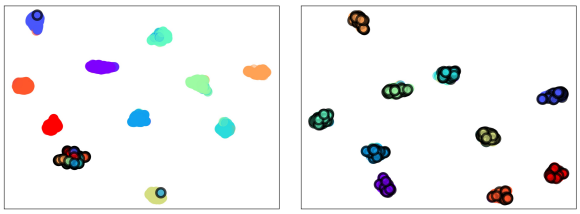


Figure 3. Visualizing the training stability. Solid and dashed lines denote the results of LP($\mathcal{D}_r$) and LP($\mathcal{D}_f$). Compared to NegGrad, which requires well-timed early stopping, our framework converges to a stable point.

our framework is the most effective at both countering the backdoor attacks and maintaining the model performance on clean samples. In contrast, although FT and NegGrad are both relatively successful in reducing the attack success rate, they compromise the overall model performance. Figure 4 visualizes the feature representations of the backdoor attacks, before and after unlearning has taken place. In this figure, points outlined in black represent the samples that have been poisoned by a backdoor trigger. Figure 4(a) demonstrates that these poisoned samples exhibit a shared semantic in the feature representation space. Our framework successfully eradicates the information pertaining to backdoor trigger, as shown in Figure 4(b), purging the backdoor effects from the model.

Training stability To verify the stability of our framework during training, we evaluate LP($\mathcal{D}_r$) and LP($\mathcal{D}_f$) as training progresses, and compare these results with those from



(a) Original model (b) Ours

Figure 4. UMAP visualization of CIFAR-10 train set under a backdoor attack scenario, (a) before unlearning and (b) after unlearning with our framework. Data points with black edges indicate the forget samples, which are poisoned by a backdoor trigger.

NegGrad [33]. Figure 3 shows that the model unlearning with our framework quickly converges to the desired performance in terms of LP($\mathcal{D}_r$) and LP($\mathcal{D}_f$), and maintains this performance even with excess training. On the other hand, the model unlearned with NegGrad experiences quick and sudden changes in LP($\mathcal{D}_r$) and LP($\mathcal{D}_f$), diverging to suboptimal performance with more iterations. Thus, the timing of early stopping is imperative to obtain an unlearned model with reasonable performance, making it more difficult to use NegGrad in practical scenarios.

6. Conclusion

We presented a novel machine unlearning framework by leveraging feature representations. We began by developing a novel metric, dubbed as dimensional alignment, which analyzes the unlearning in latent feature spaces by measuring the alignment between eigenspaces of the forget and retain sets. This metric serves as both a robust analytical tool and a powerful objective for guiding the unlearning process. Our holistic unlearning framework integrates dimensional alignment, self-distillation, and an alternate training scheme to facilitate effective and stable unlearning. Finally, we highlighted the limitations of established evaluation metrics for machine unlearning and introduced new feature-level evaluation metrics that more accurately reflect the goals of machine unlearning. We believe these contributions advance our understanding and assessment of unlearning algorithms, moving the field toward more reliable, effective, and transparent unlearning practices in machine learning systems.

Acknowledgements This work was partly supported by Samsung Advanced Institute of Technology (SAIT), and by the National Research Foundation of Korea grant [No.2022R1A2C3012210] and the Institute of Information communications Technology Planning & Evaluation (IITP) grants [No.RS-2022-II220959, No.RS-2021-II211343, No.RS-2021-II212068], funded by the Korean government (MSIT).

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. 5

[2] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE SP*, 2021. 11

[3] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE SP*, 2015. 11

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Rafal Józefowicz, Armand Joulin, Piotr Bojanowski, and Matthijs Douze. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 5

[5] Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *AAAI*, 2024. 1

[6] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *AAAI*, 2023. 1, 11

[7] Cynthia Dwork. Differential privacy. In *ICALP*, 2006. 11

[8] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *AAAI*, 2024. 6

[9] Shashwat Goel, Ameya Prabhu, and Ponnurangam Kumaraguru. Evaluating inexact unlearning requires revisiting forgetting. *CoRR abs/2201.06640*, 2022. 5, 12

[10] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *CVPR*, 2020. 5, 11

[11] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *ECCV*, 2020. 11, 12

[12] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI*, 2021. 1

[13] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 7

[14] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *ICML*, 2020. 11

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 5

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 5

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS*, 2014. 4

[18] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NIPS*, 2018. 11

[19] Dongwan Kim and Bohyung Han. On the stability-plasticity dilemma of class-incremental learning. In *CVPR*, 2023. 5

[20] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017. 11

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[22] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *NeurIPS*, 2023. 1, 4, 5, 11

[23] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[24] Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, PRANAY SHARMA, Sijia Liu, et al. Model sparsity can simplify machine unlearning. In *NeurIPS*, 2023. 7, 11

[25] Xiaolong Ma, Geng Yuan, Xuan Shen, Tianlong Chen, Xuxi Chen, Xiaohan Chen, Ning Liu, Minghai Qin, Sijia Liu, Zhangyang Wang, et al. Sanity checks for lottery tickets: Does your winning ticket really win the jackpot? In *NeurIPS*, 2021. 11

[26] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 3

[27] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, pages 931–962. PMLR, 2021. 11

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[29] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017. 13

[30] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *EUSIPCO*, 2007. 3

[31] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE SP*. 4

[32] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In *ACM SIGSAC*, 2019. 13

[33] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *EuroS&P*, 2022. 1, 4, 5, 8, 11

[34] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*, 2021. 5

[35] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 5

[36] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE CSF*, 2018. 13

[37] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 1

[38] Zijie Zhang, Yang Zhou, Xin Zhao, Tianshi Che, and Lingjuan Lyu. Prompt certified machine unlearning with randomized gradient smoothing and quantization. In *NeurIPS*, 2022. 11