Decoupled Distillation to Erase: A General Unlearning Method for Any Class-centric Tasks

Yu Zhou[1,*], Dian Zheng[1,*], Qijie Mo[1], Renjie Lu[1], Kun-Yu Lin[1,†], Wei-Shi Zheng[1,2,3,4,†]
[1]School of Computer Science and Engineering, Sun Yat-sen University, China;
[2]Peng Cheng Laboratory, China;
[3]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China;
[4]Guangdong Province Key Laboratory of Information Security Technology, China;
zhouy635@mail2.sysu.edu.cn, kunyulin14@outlook.com, wszheng@ieee.org

Abstract

*In this work, we present **DE**coup**LE**d Distillation **T**o **E**rase (**DELETE**), a general and strong unlearning method for any class-centric tasks. To derive this, we first propose a theoretical framework to analyze the general form of unlearning loss and decompose it into forgetting and retention terms. Through the theoretical framework, we point out that a class of previous methods could be mainly formulated as a loss that implicitly optimizes the forgetting term while lacking supervision for the retention term, disturbing the distribution of pre-trained model and struggling to adequately preserve knowledge of the remaining classes. To address it, we refine the retention term using "dark knowledge" and propose a mask distillation unlearning method. By applying a mask to separate forgetting logits from retention logits, our approach optimizes both the forgetting and refined retention components simultaneously, retaining knowledge of the remaining classes while ensuring thorough forgetting of the target class. Without access to the remaining data or intervention (*i.e.*, used in some works), we achieve state-of-the-art performance across various benchmarks. What's more, DELETE is a general solution that can be applied to various downstream tasks, including face recognition, backdoor defense, and semantic segmentation with great performance.*

1. Introduction

The growing importance of "the right to be forgotten" [53] highlights concerns around pretrained deep models, which heavily rely on user data and large-scale, unfiltered web-crawled datasets [12, 41, 42]. Such reliance may compromise both user privacy and model reliability [16, 41, 44]. How to erase the influence exerted by specific subsets of training data, known as "machine unlearning" [3], has become a pressing issue. To circumvent the high computational expense of retraining [21], there is a need for more efficient unlearning methods. In this work, we focus on class-centric unlearning, which is important for applications like facial recognition, backdoor defense, and semantic segmentation, where forgetting specific classes is needed [9, 30].

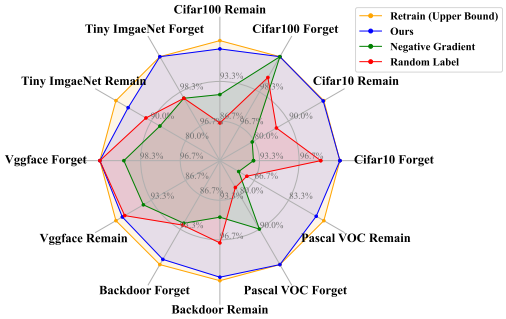


Figure 1. Performance comparison of our method with previous works and retraining (*i.e.*, the upper bound set to 100%) across image classification, data poisoning, face recognition and semantic segmentation, evaluated on forgetting and remaining performance.

In this work, we aim to explore a general unlearning solution for any class-centric tasks, which could not be implemented by existing works as shown in Fig 1. To achieve it, we first propose a theoretical framework to derive the general form of unlearning loss and decouple the unlearning loss into two components: forgetting loss and retention loss, which serve as optimization objectives for forgetting and knowledge retention, respectively. Based on our theoretical framework, we reformulate previous re-label based methods [8, 9, 19] (*i.e.*, a prominent category of unlearning techniques that achieve unlearning by assigning incorrect labels to forget samples.) and observe that they implicitly optimize the forgetting term but lack explicit supervision over the retention term. This limitation hinders the ability to effectively retain knowledge of remaining classes and therefore disturbs the distribution of pre-trained model [9, 14]. To address this, we further refine the retention loss by leveraging "dark knowledge" [61], *i.e.*, the relative magnitudes of the probabilities for the remaining classes. By apply-

*These authors contributed equally.
†These authors are corresponding authors.

ing a masking strategy that separates forgetting logits from retention logits, we develop a post-hoc decoupled distillation unlearning method. This approach simultaneously optimizes both the forgetting and refined retention loss components, enabling effective unlearning while retaining essential knowledge. Combining the theoretical framework with our novel unlearning loss, we propose the Decoupled Distillation to Erase method, termed DELETE.

By only modifying the unlearning loss, without bells and whistles we outperform existing unlearning methods by a large margin across different benchmarks, model architectures, and forgetting settings. Additionally, we apply our unlearning loss to downstream tasks such as face recognition, data poisoning and semantic segmentation, and achieve great performance success both in forgetting the target class and remaining others as shown in Fig 1, highlighting its strong generalization capabilities and promising applications in real-world scenarios. To the best of our knowledge, this work presents the first comprehensive and fair comparison of all existing class-centric unlearning methods that do not rely on remaining data or intervention, evaluated under thorough settings across a wide range of benchmarks.

In summary, our main contributions are as follows:

- We adopt a novel theoretical analysis that decomposes the unlearning loss into a forgetting term and a retention term. This inspires a new perspective on leveraging and preserving knowledge within the model based on logits. Under the analysis framework, we also point out that some previous methods implicitly adapt the forgetting term but lack supervision for the retention loss.
- Motivated by the analysis, we further refine the retention loss using "dark knowledge" and introduce the post-hoc decoupled distillation unlearning method, termed DELETE. By masking to decouple the logits of forgotten and non-forgotten classes, we optimize both the decoupled objectives simultaneously, which leads to both effective unlearning and excellent knowledge preservation.
- Through extensive experiments across a wide range of benchmarks, we demonstrate our method's SOTA performance. Additionally, we apply it to multiple real-world downstream tasks, such as face recognition, backdoor defense, and semantic segmentation, highlighting its strong generalization capabilities.

2. Related Work

Machine Unlearning. Machine Unlearning is the task of removing the influence of specific data from a pretrained model [3, 7, 18]. This task addresses the "right to be forgotten" [53], which is crucial for privacy protection [3] and enhancing AI safety [17, 30, 37]. Retraining from scratch is a common yet naive approach, as it incurs substantial computational overhead [21, 58]. Some methods explore machine unlearning for traditional machine learning tasks, such as linear regression [2, 34], k-means [18], random forests [4], and SVM [10]. However, these methods typically leverage the convex nature of the problem or other theoretical constraints, so that can't be adopted on DNN [2, 6, 7, 32, 33, 56, 57, 59, 63, 64].

Unlearning in DNN. The urgent need for fast unlearning methods in DNNs leads to various approaches. Some methods require intervention during the pre-training process. SISA [3] splits the model into multiple sub-models trained independently, and selectively removes these sub-models to unlearn. Unrolling SGD [50] adds extra loss terms during training, while other methods [21, 58] rely on stored gradient information to facilitate and speed up unlearning. However, these designs increase computational overhead, reduce efficiency, and may degrade model performance [36, 60], making them impractical for machine-as-a-service (MaaS) and complex real-world applications [22, 55].

To achieve effective forgetting while preserving unrelated knowledge, other methods attempt to use the remaining data. Fisher Forgetting [19] calculates the Fisher Information Matrix based on remaining data, adding noise to scrub parameters associated with forgetting data. SSD [15] calculates parameter importance for both forgotten and remaining data, then applies weight perturbations to unlearn. Additionally, some methods introduce regularization terms or an extra repair stage to reduce interference with unrelated knowledge [27, 49]. However, in real-world applications, factors such as restricted data access, data expiration policies, storage costs, or privacy concerns may prevent providers from storing remaining data [1, 8, 29]. In order to address these challenges, unlearning should ideally be performed without relying on such data. In these cases, the above methods become unusable or suffer from observable performance degradation [15, 45, 49].

Therefore, a stricter and more practical machine unlearning scenario needs to be explored. Our method does not interfere with the pre-training process, or rely on remaining data, making it readily applicable in real-world scenarios.

Knowledge Distillation. Knowledge distillation [24], initially proposed by Hinton et al., is a technique that transfers knowledge from a teacher model to a student model. FitNets [38] further advances this approach by incorporating intermediate features to better align teacher and student models. Decoupled knowledge distillation [61] separate the logits into TCKD and NCKD components, enhancing model capability and generalization.

In recent years, knowledge distillation has been applied across various vision tasks, including machine unlearning [20, 35, 39, 40, 48, 54]. Bad Teacher [11] initializes a pair of competent and incompetent teacher models, using knowledge distillation to selectively unlearn while retaining unrelated knowledge. SCRUB [27] leverages the teacher model's outputs on remaining data to preserve knowledge.

Our approach decouples the unlearning loss into two distinct components and incorporates mask to guide the distillation process. Unlike previous methods, we achieve selective forgetting without access to remaining data.

3. Problem Definition for Machine Unlearning

We now define the notion of class-centric machine unlearning. Let the dataset be denoted as $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N} \subseteq \mathcal{X} \times \mathcal{Y}$, consisting of N data points. $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ represents the input image, and $y \in \mathcal{Y} = \{1, \cdots, K\}$ is the ground truth label of the image. Let the original classification model trained on the dataset $\mathcal{D}$ be denoted as $f_{\theta_o} : \mathcal{X} \to \mathcal{Y}$, where θ_o represents the model parameters.

In machine unlearning, the original dataset is divided into two non-overlapping subsets, $\mathcal{D}_f$ and $\mathcal{D}_r$. The goal of machine unlearning is to erase the knowledge related to $\mathcal{D}_f$ from the model while preserving knowledge related to $\mathcal{D}_r$ remains unaffected. Some existing methods use all or part of $\mathcal{D}_r$ during unlearning to preserve the model's related knowledge. However, due to privacy concerns and restricted data access in real-world applications [1, 8, 29], we adopt a stricter machine unlearning setting: during unlearning, **only the data from $\mathcal{D}_f$ can be used.**

As a consequence, the unlearning algorithm trains the model parameters θ_o on $\mathcal{D}_f$ to obtain new parameters θ. The unlearned model f_θ is required to exhibit incorrect classification on $\mathcal{D}_f$, while maintaining high classification accuracy on $\mathcal{D}_r$. The model f_{θ^*}, retrained on $\mathcal{D}_r$, is typically regarded as the upper bound for machine unlearning [9, 51]. The output of the unlearned model f_θ should be as close as possible to this target model f_{θ^*}.

4. Theoretical Analysis and Method

In this section, we first present a theoretical analysis framework to explore the general form of unlearning loss and decompose it into forgetting and retention losses. Based on the theoretical analysis, we show that re-label unlearning methods are a special case of unlearning loss. Then we suggest a forgetting condition on the forgetting loss derived from experimental observations; while re-label methods also implicitly satisfy this, they lack supervision on the remaining classes. Therefore, we introduce a retention condition for the retention loss. By combining both conditions, we finally propose a mask distillation unlearning method.

4.1. Theoretical Analysis

1) Decoupling Unlearning Loss. In this section, inspired by DKD [61], we propose the unlearning loss and decompose it into two parts: the **forgetting loss** and the **retention loss**. For a classification sample $\mathbf{x} \in \mathcal{D}_f$ from class u to be unlearned, let the target probability distribution output be $\mathbf{p} \in \mathbb{R}^K$, and the output of the actual unlearning model f_θ be $\mathbf{q} \in \mathbb{R}^K$. Typically, the unlearning loss should reduce KL divergence, given by $\mathcal{L} = \mathrm{KL}(\mathbf{p} \| \mathbf{q})$.

Let p_i denote the i-th component of the probability distribution for the target $\mathbf{p}$. Specifically, p_u and $p_{\setminus u} = \sum_{i \neq u} p_i$ denote the probabilities corresponding to the forgetting class and the sum for all other classes, respectively. We denote the binary-class probabilities as $\mathbf{p}^{(b)} = [p_u, p_{\setminus u}]$. Similarly, the predicted probabilities q_u, $q_{\setminus u}$ and the binary-class distribution $\mathbf{q}^{(b)} = [q_u, q_{\setminus u}]$ are defined. For all classes $i \neq u$, we define $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$ as the re-normalized probability distributions over these $K-1$ classes, *i.e.*, each element satisfies $\hat{p}_i = \frac{p_i}{\sum_{i \neq u} p_i}$ and $\hat{q}_i = \frac{q_i}{\sum_{i \neq u} q_i}$.

The unlearning loss function, which reduces the gap between the model's predicted distribution and the target distribution, can be decomposed into two parts:

$$\mathcal{L} = p_u \log \left(\frac{p_u}{q_u} \right) + \sum_{i \neq u} p_i \log \left(\frac{p_i}{q_i} \right). \tag{1}$$

For the second term, leveraging the identities $p_i = \hat{p}_i \cdot p_{\setminus u}$ and $q_i = \hat{q}_i \cdot q_{\setminus u}$ for $i \neq u$, we have:

$$\begin{aligned} \sum_{i \neq u} p_i \log \left(\frac{p_i}{q_i} \right) &= p_{\setminus u} \sum_{i \neq u} \hat{p}_i \log \left(\frac{\hat{p}_i \cdot p_{\setminus u}}{\hat{q}_i \cdot q_{\setminus u}} \right) \\ &= p_{\setminus u} \sum_{i \neq u} \hat{p}_i \left(\log \left(\frac{\hat{p}_i}{\hat{q}_i} \right) + \log \left(\frac{p_{\setminus u}}{q_{\setminus u}} \right) \right) \\ &= p_{\setminus u} \log \left(\frac{p_{\setminus u}}{q_{\setminus u}} \right) + p_{\setminus u} \sum_{i \neq u} \hat{p}_i \log \left(\frac{\hat{p}_i}{\hat{q}_i} \right). \end{aligned}$$

Substituting this result into the equation Eq. (1), the loss function becomes:

$$\begin{aligned} \mathcal{L} &= p_u \log \left(\frac{p_u}{q_u} \right) + p_{\setminus u} \log \left(\frac{p_{\setminus u}}{q_{\setminus u}} \right) + p_{\setminus u} \sum_{i \neq u} \hat{p}_i \log \left(\frac{\hat{p}_i}{\hat{q}_i} \right) \\ &= \mathrm{KL}(\mathbf{p}^{(b)} \| \mathbf{q}^{(b)}) + p_{\setminus u} \mathrm{KL}(\hat{\mathbf{p}} \| \hat{\mathbf{q}}). \end{aligned} \tag{2}$$

Thus, we decouple the original loss function into two components: one focused on enforcing forgetting of the target class, and the other supervising the remaining classes. The former arises from the fact that $\mathbf{p}^{(b)} = [p_u, p_{\setminus u}] = [p_u, 1 - p_u]$ depends only on the probability of the forgotten class, p_u, and similarly, $\mathbf{q}^{(b)}$ depends on q_u. We refer to these components as the **forgetting loss** and **retention loss**.

2) Rethinking Re-label Unlearning Method. In this part, we demonstrate that re-label unlearning methods, a common approach within unlearning tasks, represent a special case of the previously established theoretical framework. This allows us to leverage the conclusions drawn in Eq. (2).

Re-label methods assign a fixed incorrect label $r \neq u$ to sample $\mathbf{x}$, as a form of supervised unlearning. For example, random label assigns a randomly selected label to $\mathbf{x}$ [19]. Some methods use adversarial attacks to derive replacement labels, by attacking $\mathbf{x}$ against the frozen model f_{θ_o} [8, 9]. The unlearning loss $\mathcal{L}_{\mathrm{re-label}}$ is defined as the cross-entropy

loss between $\mathbf{x}$ and the incorrect label r, formulated as:

$$\arg\min_{\theta} \mathcal{L}_{\mathrm{re-label}} = \arg\min_{\theta} -\log(q_r). \tag{3}$$

We then demonstrate that this is equivalent to the KL-divergence between the one-hot vector of class r and the model's output. Let the one-hot vector for class r be denoted as $\mathbf{e}^r$. When the target output distribution $\mathbf{p} := \mathbf{e}^r$, note the fact that target $\mathbf{p}$ is independent of the optimization parameter θ and the properties of one-hot vectors. In this case, we have:

$$\begin{aligned}\arg\min_{\theta} \mathcal{L}_{\mathrm{re-label}} &= \arg\min_{\theta} -\log(q_r) + \log(p_r) \\ &= \arg\min_{\theta} \log\left(\frac{p_r}{q_r}\right) \\ &= \arg\min_{\theta} \sum_{i=1}^{K} p_i \log\left(\frac{p_i}{q_i}\right).\end{aligned}$$

By transforming the re-label unlearning loss Eq. (3) into a KL-divergence form as in Eq. (1), we enable further discussion within the decomposition framework introduced earlier. In other words, re-label methods represent a special case of Eq. (2), where the target distribution is set as a one-hot vector.

3) Implicit Forgetting Condition in Re-labeling. While the target distribution is not accessible during unlearning, we can approximate $\mathbf{p}$ by observing the probability distributions generated by the retrain model, which is typically considered the gold standard for machine unlearning [9, 51]. As shown in the Tab. 1 in the subsequent experiments, after unlearning, the probability of the forgotten class p_u approaches 0. Previous re-label methods implicitly apply this approximation by using a one-hot target, where p_u is set as $p_u := e_u = 0$.

Based on these observations, we formally adopt the **forgetting condition** $p_u = 0$ in **forgetting loss** from Eq. (2), as an approximation of target distribution. Consequently, we obtain $p_{\setminus u} = 1 - p_u = 1$, leading to $\mathbf{p}^{(b)} = [p_u, p_{\setminus u}] = [0, 1]$, which we denote as $\mathbf{e}^{(b)} = [0, 1]$. The loss function in Eq. (2) can be further expressed as:

$$\begin{aligned}\mathcal{L} &= \mathrm{KL}(\mathbf{e}^{(b)} \| \mathbf{q}^{(b)}) + p_{\setminus u} \mathrm{KL}(\hat{\mathbf{p}} \| \hat{\mathbf{q}}) \\ &= \mathrm{KL}(\mathbf{e}^{(b)} \| \mathbf{q}^{(b)}) + p_{\setminus u} \sum_{i \neq u} \hat{p}_i \log\left(\frac{\hat{p}_i}{\hat{q}_i}\right) \\ &= \mathrm{KL}(\mathbf{e}^{(b)} \| \mathbf{q}^{(b)}) + p_{\setminus u} \hat{p}_r \log\left(\frac{\hat{p}_r}{\hat{q}_r}\right) + p_{\setminus u} \sum_{i \neq u, r} \hat{p}_i \log\left(\frac{\hat{p}_i}{\hat{q}_i}\right).\end{aligned} \tag{4}$$

It suggests that the loss function should not only enforce forgetting the target class via the first term, but also supervise predictions for remaining classes via the second and third terms. This structure helps maintain the model's predictions for other classes before and after unlearning, preventing excessive forgetting.

However, re-label unlearning methods lack this supervision for unrelated classes $i \neq u, r$. Recall re-label methods are equivalent to using a one-hot vector for supervision. The corresponding element in the target distribution is assigned $p_i := 0$, (*i.e.*, $\hat{p}_i := 0$), transforming Eq. (4) into:

$$\mathcal{L}_{\text{re-label}} = \mathrm{KL}(\mathbf{e}^{(b)} \| \mathbf{q}^{(b)}) + p_{\setminus u} \hat{p}_r \log\left(\frac{\hat{p}_r}{\hat{q}_r}\right).$$

As a result, the model's prediction for other classes is left unsupervised, which would otherwise serve as the third term in Eq. (4). This indicates a lack of supervision for the predictions of non-forgetting classes in the **retention loss**. Thus, some methods leverage remaining data to preserve performance on retained classes [14].

4.2. Mask Distillation Method

Previous work neglected the supervision of remaining classes, hindering the preservation of predictions for other classes. However, when forgetting specified classes, it is important to keep the predictions for other classes unchanged [9]. Additionally, preserving the relative magnitudes of the model's outputs for other classes, which reflect its "dark knowledge", also plays a crucial role [61]. Thus, we aim to further retain the relative magnitudes of the non-forgotten outputs on **retention loss**, under the forgetting condition. This motivates the use of distillation based on the original model's logits, rather than one-hot supervision, which tends to be overly rigid.

To summarize, our goal is to approximate the target distribution using three **key observations and conditions**, building further on the Sec. 4.1:

- **forgetting condition**: The probability output for the unlearning class, p_u, approaches zero , *i.e.*, $p_u = 0$, as discussed in Sec. 4.1.
- **retaining condition**: The relative magnitude for the remaining classes remain unchanged, *i.e.*, for $i \neq u$, $p_i \propto \mathrm{Softmax}(f_{\theta_o}(\mathbf{x}))_i$.
- **probability property**: The target probability distribution must satisfy the basic property of a probability distribution, *i.e.*, $\sum_{i=1}^{k} p_i = 1$.

This inspires us to supervise the current unlearn model f_θ leveraging the variant of distribution predicted by the frozen original model f_{θ_o}. By applying a mask and vector normalization to the model output, we obtain an approximate target distribution that satisfies the above conditions.

Let $\mathbf{z}(\mathbf{x}; \theta_o) = f_{\theta_o}(\mathbf{x})$ (simplified as $\mathbf{z}$ hereafter) represent the logits of the frozen original model for $\mathbf{x}$, where $\mathrm{Softmax}(\mathbf{z})$ represents the corresponding probability distribution. We introduce $\mathrm{Mask}_u(\cdot)$ as a masking function applied to $\mathbf{v} \in \mathbb{R}^K$, which sets the probability of class u to zero while leaving the other dimensions unchanged. Recalling that $\mathbf{e}^u$ is the u-th column of the identity matrix I, this masking function is defined as the Hadamard product:

$$\mathrm{Mask}_u(\mathbf{v}) = (\mathbf{1} - \mathbf{e}^u) \odot \mathbf{v}.$$

Algorithm 1 Pseudocode of DELETE in a PyTorch-like style.

```
# unlearn_model: the classification model to unlearn
# images: the input images to forget (N, 3, H, W)
# labels: the corresponding labels (N, K)

# Initialize the frozen model with unlearn model
frozen_model = copy(unlearn_model)

# Disable gradient computation for the frozen model
frozen_model.params.require_grad_(False)

for images, labels in loader:
   # Create a decoupled distillation mask
   mask = zeros((N, K))
   mask[arange(N), labels] = -float("inf")

   # Compute soft targets using the frozen model
   frozen_outputs = frozen_model.forward(images)
   masked_outputs = mask + frozen_outputs
   soft_targets = softmax(masked_outputs, dim=1)

   # Forward pass through the unlearn model
   logits = unlearn_model.forward(images)
   loss = KLDivLoss(softmax(logits, dim=1),
        soft_targets)

   # Update the unlearn model
   loss.backward()
   update(unlearn_model.params)
```

The $\mathrm{Normalize}(\cdot)$ function normalizes the vector to ensure that the sum of its elements equals 1, which is achieved by dividing each element by the total sum of all.

By applying the masking and normalization function to process the frozen model's output as the target, we have:

$$\mathcal{L} = \mathrm{KL}\left(\mathrm{Normalize}(\mathrm{Mask}_u(\mathrm{Softmax}(\mathbf{z})))\,\|\,\mathbf{q}\right). \quad (5)$$

The target in Eq. (5) satisfies all these **conditions**. By masking out the probability for the target class, we satisfy **forgetting condition**. The probabilities for the other classes remain after the masking process, and after normalization, they are proportional to the original model's output probabilities, satisfying **retaining condition**. Clearly, the normalize function ensures the **probability property**.

Compared to explicitly specifying a one-hot target vector, the masking design allows for a more fine-grained utilization of the logits for the decoupled forgetting and non-forgetting components. This approach not only enables effective unlearning but also preserves knowledge of the non-forgotten classes, as shown later in Sec. 5.3.

Moreover, we demonstrate that, with a carefully designed mask function $\mathrm{Mask}'_u(\cdot)$, the masking and softmax steps can be interchanged. This reordering allows the softmax vector to naturally sum to 1, reducing the need for normalization while achieving equivalent results. Further details are provided in the Appendix. The resulting loss function, derived from Eq. (5), is as follows:

$$\mathcal{L} = \mathrm{KL}\left(\mathrm{Softmax}(\mathrm{Mask}'_u(\mathbf{z}))\,\|\,\mathbf{q}\right).$$

For clarity, the Algorithm 1 provides the final method.

5. Experiments

5.1. Experimental Setup

Considering privacy requirements, restricted access, computational constraints, and other real-world limitations as mentioned before, we impose two strict constraints in our experiments: **(i)** No access to the remaining data; **(ii)** No intervention during the pre-training phase. Additional implementation details are provided in the appendix.

Datasets. To evaluate the performance on class-centric unlearning task, we use CIFAR-10, CIFAR-100 [26], and Tiny ImageNet [28] datasets. For both single-class and multi-class forgetting tasks, we randomly select a set of classes, then unlearn the same selected classes across methods.

Models. We adopt ResNet-18 [23] as the primary architecture for forgetting experiments. Additionally, we evaluate methods on ViT-s [13], Swin-T [31], and VGGNet-16 [47] to assess effectiveness across different model structures.

Baselines. In our experiments, we compare the following baselines : Random Label, Negative Gradient, Finetune, Fisher Forgetting [19], Boundary Shrink [9], Boundary Expand [9], Learn to Unlearn [8], Bad Teacher [11], Saliency Unlearn [14], SCRUB [27] and Influence Unlearn [25].

Additionally, we include the performance of the original model and the retrained model, with the latter often considered the gold standard in machine unlearning [9, 51].

Metrics. We employ the following accuracy to evaluate effectiveness:$\mathbf{Acc_f}$ on the forget training data $\mathcal{D}_\mathrm{f}$, $\mathbf{Acc_r}$ on the remain training data $\mathcal{D}_\mathrm{r}$, $\mathbf{Acc_{ft}}$ on the forget test data $\mathcal{D}_\mathrm{ft}$, and $\mathbf{Acc_{rt}}$ on the remain test data $\mathcal{D}_\mathrm{rt}$.

We further use **H-Mean** as an overall performance measure of model forgetting and retention. Similar to gs-lora [62], H-Mean is computed as the harmonic mean of Acc_rt and $\mathrm{Drop}_\mathrm{ft}$, where $\mathrm{Drop}_\mathrm{ft}$ represents the performance drop on the forget test data after unlearning.

What's more, we employ **MIA**, the membership inference attack [46] success rate, as another indicator of forgetting effectiveness. MIA measures the proportion of samples in $\mathcal{D}_\mathrm{f}$ that the unlearn model may still "memorize", revealing potential remnants of the forget data [14].

Metrics associated with Acc_f, Acc_ft, and MIA should approach zero to indicate effective forgetting, while metrics such as Acc_r, Acc_rt, and H-Mean are expected to align with original model's, demonstrating retention of knowledge in the remaining classes [9, 14, 62].

5.2. Results and Comparisons

Performance on Different Datasets. In this part, we discuss the single-class forgetting performance across various datasets. Although our experimental setup does not allow access to remaining data or intervention, we still compare several methods that have these, marking them with gray.

Using CIFAR-10 as an example in Tab. 1, although all methods achieve varying degrees of forgetting in Acc_ft,

Table 1. Performance comparison of single-class forgetting across different unlearning methods on CIFAR-10 and CIFAR-100 datasets. Gray indicates methods with remain data or intervention, and **bold** indicates the single best result among methods w/o remain data or intervention (if multiple, results are not in bold). The same notation applies hereafter.

Method	CIFAR-10						CIFAR-100					
	$\text{Acc}_\text{f}\downarrow$	$\text{Acc}_\text{r}\uparrow$	$\text{Acc}_\text{ft}\downarrow$	$\text{Acc}_\text{rt}\uparrow$	H-Mean $\uparrow$	MIA $\downarrow$	$\text{Acc}_\text{f}\downarrow$	$\text{Acc}_\text{r}\uparrow$	$\text{Acc}_\text{ft}\downarrow$	$\text{Acc}_\text{rt}\uparrow$	H-Mean $\uparrow$	MIA $\downarrow$
Original Model	100	100	97.00	95.31	-	99.08	100	99.98	67.00	77.79	-	98.00
Retrain Model	0	100	0	95.20	96.09	0	0	99.97	0	77.64	71.93	0
Random Label	0.10	90.66	1.60	82.18	88.30	0	1.20	96.60	1.00	66.99	66.49	0
Negative Gradient	11.12	83.94	7.20	75.52	82.04	7.44	2.20	97.80	0	70.66	68.78	1.20
Boundary Shrink	0.28	90.34	4.50	84.56	88.35	0.22	1.40	96.48	0	68.46	67.72	0
Boundary Expand	11.32	90.74	15.00	79.37	80.66	0.86	2.40	98.59	1.00	71.69	68.73	0
Influence Unlearn	1.04	99.58	0.60	93.21	94.78	0	0	97.83	0	71.03	68.96	0
Learn to Unlearn	12.36	96.49	8.60	90.02	89.20	7.56	1.60	97.70	0	70.29	68.61	0.40
Bad Teacher	1.98	92.23	18.80	87.13	82.42	0.52	1.60	98.46	3.00	71.38	67.49	0.20
Bad Teacher	0.34	99.99	0.30	94.98	95.83	0	0.40	99.98	0	77.63	71.92	0
Saliency Unlearn	9.98	79.19	10.50	70.18	80.53	11.70	1.20	88.07	3.00	59.73	61.79	1.00
Saliency Unlearn	0.84	99.73	0.10	93.91	95.38	0.02	0.80	99.21	0	74.27	70.45	0
Fisher Forget	68.94	57.33	68.30	59.73	38.77	59.50	66.00	74.52	37.00	54.68	38.74	68.60
SCRUB	0	99.92	0	94.94	95.96	0	0	99.95	0	77.41	71.83	0
Finetune	0.12	99.53	0	93.83	95.39	0	0	99.30	0	74.15	70.39	0
Ours	**0**	**100**	**0**	**95.03**	**96.00**	0	0	**99.97**	0	**76.57**	**71.47**	0

Table 2. Performance comparison across different models.

	$\text{Acc}_\text{f}\downarrow$	$\text{Acc}_\text{r}\uparrow$	$\text{Acc}_\text{ft}\downarrow$	$\text{Acc}_\text{rt}\uparrow$	H-Mean $\uparrow$	MIA $\downarrow$
VGGNet-16						
Original Model	99.94	99.94	91.20	92.02	-	99.88
Retrain Model	0	99.71	0	92.49	91.84	0
Boundary Shrink	6.68	91.15	6.80	82.79	83.59	2.84
Ours	0	99.66	0	92.08	91.64	0
Swin-T						
Original Model	85.00	86.58	82.70	82.68	-	82.94
Retrain Model	0	86.41	0	82.77	82.73	0
Boundary Shrink	5.54	62.94	2.50	58.18	67.44	11.36
Ours	0	85.37	0	83.09	82.89	0
ViT-S						
Original Model	90.50	90.57	72.70	75.73	-	84.60
Retrain Model	0	92.42	0	76.44	74.52	0
Boundary Shrink	1.84	73.16	1.70	65.59	68.19	1.66
Ours	0	90.74	0	77.21	74.89	0

Table 3. Performance comparison with frozen linear classifier.

Method	$\text{Acc}_\text{f}\downarrow$	$\text{Acc}_\text{r}\uparrow$	$\text{Acc}_\text{ft}\downarrow$	$\text{Acc}_\text{rt}\uparrow$	H-Mean $\uparrow$	MIA $\downarrow$
Original Model	100	100	97.00	95.31	-	99.08
Retrain Model	0	100	0	95.12	96.05	0
Random Label	0.82	93.60	6.90	85.51	87.75	0
Negative Gradient	19.20	87.83	15.00	79.37	80.66	13.88
Boundary Shrink	1.12	87.97	9.30	82.43	84.98	1.36
Boundary Expand	58.44	32.02	69.60	39.13	32.23	66.62
Bad Teacher	4.14	86.26	24.80	81.51	76.57	2.14
Saliency Unlearn	10.58	85.95	11.20	76.14	80.68	3.78
Influence Unlearn	0.40	99.61	0.20	93.59	95.17	0.20
Learn to Unlearn	21.00	97.86	17.60	91.09	84.84	15.54
Ours	**0**	**100**	**0**	**95.44**	**96.21**	0

some methods like Negative Gradient maintain Acc_f and Acc_ft around 10%, indicating incomplete forgetting. In contrast, our method achieves complete forgetting in Acc_ft while maintaining an excellent Acc_rt, with a minimal gap of only 0.17% compared to the retrain model. When experiments are extended to larger datasets such as CIFAR-100 and Tiny ImageNet (Tab. 1 and Appendix), the performance of some methods fluctuates significantly. Notably, our method consistently outperforms all other methods across all datasets and metrics.

Performance on Different Models. To evaluate effectiveness across various models, we further conduct single-class unlearning experiments with VGGNet-16, Swin-T, and Vit-S, as shown in Tab. 2. A full comparison of methods is provided in the Appendix. At the same performance of Acc_ft, our method shows a maximum performance gap of only 0.77% in Acc_rt compared to the retrain model, demonstrating consistently excellent results across different architectures while Boundary Shrink shows limited performance, validating the robustness of DELETE.

Performance with Frozen Linear Classifier. A key concern in machine unlearning is whether knowledge of the target class is truly forgotten. Studies suggest that forgetting often occurs in the model's linear classifier [8]. However, merely modifying classifier to misclassify without erasing knowledge at the feature level doesn't fulfill the goal of machine unlearning [19]. Incomplete forgetting within the feature extractor could also introduce risks of privacy leakage.

To address this, we freeze the model's linear classifier and train only the feature extractor to forget. Results, shown in Tab. 3, demonstrate that our method consistently outperforms existing approaches across all metrics. Boundary Expand shows a significant drop in overall performance, suggesting such methods **achieve unlearning primarily at the classifier level rather than at the feature level**.

Performance on Multi-Class Forgetting. Building on single-class forgetting in CIFAR-100, we further experiment with forgetting multiple classes. As shown in Tab. 4, the results indicate that some methods suffer significant performance drops as forgetting class number increases. When forgetting 20 classes, none of the other methods achieve both an Acc_ft of 0% and an Acc_rt above 65%, as they do in single-class forgetting. In contrast, our method demonstrates exceptional multi-class forgetting capability, showing the power of our novel loss function.

Performance Comparison of Stability and Insensitivity. In this part, we discuss the stability and learning rate in-

Table 4. Comparison of the classification accuracy of multi-class forgetting across methods on CIFAR-100 dataset.

Method	1 Class			2 Classes			5 Classes			10 Classes			20 Classes		
	$\text{Acc}_{\text{ft}}\downarrow$	$\text{Acc}_{\text{rt}}\uparrow$	H-Mean $\uparrow$	$\text{Acc}_{\text{ft}}\downarrow$	$\text{Acc}_{\text{rt}}\uparrow$	H-Mean $\uparrow$	$\text{Acc}_{\text{ft}}\downarrow$	$\text{Acc}_{\text{rt}}\uparrow$	H-Mean $\uparrow$	$\text{Acc}_{\text{ft}}\downarrow$	$\text{Acc}_{\text{rt}}\uparrow$	H-Mean $\uparrow$	$\text{Acc}_{\text{ft}}\downarrow$	$\text{Acc}_{\text{rt}}\uparrow$	H-Mean $\uparrow$
Original Model	67.00	77.79	-	78.00	77.66	-	76.80	77.72	-	78.80	77.54	-	77.40	77.74	-
Retrain Model	0	77.64	71.93	0	78.09	78.04	0	78.63	77.70	0	79.08	78.94	0	80.35	78.85
Random Label	1.00	66.99	66.49	4.00	65.97	69.75	2.80	63.51	68.35	3.10	59.97	66.92	4.45	59.35	65.45
Negative Gradient	0	70.66	68.78	1.50	69.96	73.08	0.20	68.97	72.59	2.10	67.11	71.59	3.10	40.60	52.51
Boundary Shrink	0	68.46	67.72	1.00	60.10	67.51	0.40	63.68	69.46	0.50	59.19	67.42	3.90	61.90	67.20
Boundary Expand	1.00	71.69	68.73	0.50	69.00	73.00	2.80	69.03	71.43	2.70	64.79	69.99	5.80	62.59	66.79
Influence Unlearn	0	71.03	68.96	0	67.84	72.57	2.20	62.33	67.92	**0**	37.56	50.87	20.20	43.77	49.59
Learn to Unlearn	0	70.29	68.61	0.50	69.35	73.20	0	58.80	66.61	1.80	66.56	71.40	20.65	64.90	60.55
Bad Teacher	3.00	71.38	67.49	0	59.48	67.49	0.80	62.61	68.66	0.50	58.17	66.75	3.10	55.07	63.26
Saliency Unlearn	3.00	59.73	61.79	2.50	56.60	64.70	3.00	64.64	68.92	3.20	53.84	62.89	6.05	62.31	66.52
Ours	0	**76.57**	**71.47**	0	**76.78**	**77.39**	0	**77.76**	**77.28**	0.50	**77.61**	**77.95**	**1.20**	**79.30**	**77.72**

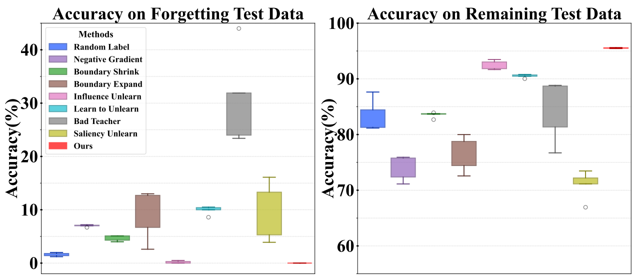


Figure 2. Boxplot comparison of accuracy on $\mathcal{D}_{\text{ft}}$ (**left**) and $\mathcal{D}_{\text{rt}}$ (**right**) across 5 runs, illustrating accuracy distributions and stability. Ours consistently achieves the lowest Acc_{ft} and the highest Acc_{rt}, demonstrating stable performance across executions.

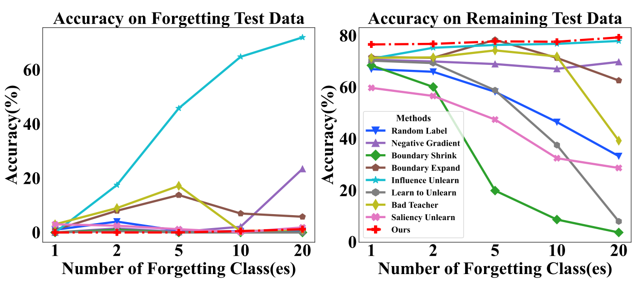


Figure 3. Accuracy comparison on $\mathcal{D}_{\text{ft}}$ (**left**) and $\mathcal{D}_{\text{rt}}$ (**right**) across multiple forgetting classes with a fixed learning rate. Ours achieves superior Acc_{ft} and Acc_{rt}, whereas all other methods exhibit varying degrees of performance degradation.

sensitivity of different methods. To evaluate the stability of each method, we conduct five repeated single-class forgetting runs. As shown in Fig. 2, some methods exhibit considerable performance variance, indicated by the width of the boxplots. Meanwhile, ours demonstrate excellent stability.

Moreover, an promising method is expected to demonstrate insensitivity to the learning rate across varying numbers of forgetting classes, *i.e.*, achieving strong performance with a fixed learning rate. Otherwise, a different number of unlearning classes would require frequent adjustments, resulting in additional computational overhead and unstable performance. As shown in Fig. 3, our method achieves effective forgetting while preserving high accuracy on the remaining data. In contrast, all other methods exhibit varying degrees of performance degradation in Acc_{rt} or Acc_{ft}, resulting in a noticeable performance gap compared to ours.

Table 5. Ablation study on forgetting and retention conditions, with varying values of α and T respectively.

	$\text{Acc}_{\text{f}}\downarrow$	$\text{Acc}_{\text{r}}\uparrow$	$\text{Acc}_{\text{ft}}\downarrow$	$\text{Acc}_{\text{rt}}\uparrow$	H-Mean $\uparrow$
Retrain Model	0	100	0	95.20	96.09
Our Method	0	100	0	95.03	96.00
Effect of varying α					
$\alpha = 0.25$	41.00	100	37.30	95.49	73.47
$\alpha = 0.50$	91.84	99.99	76.80	95.31	33.33
$\alpha = 0.75$	99.88	99.97	95.00	94.36	3.92
Effect of varying T					
$T = 5$	0	78.60	0	76.92	85.80
$T = 10$	0	30.13	0	30.38	46.27
$T = 15$	0	17.51	0	18.81	31.51

5.3. Ablation Study

In this part, we conduct ablation studies to validate the effectiveness of the forgetting and retention conditions.

For the forgetting condition, the output on forget class is required to approach 0, enforcing the erasure of target class. To validate this, we define $p_u = \alpha \cdot \text{Softmax}(f_{\theta_o}(\mathbf{x}))_u$, where setting $\alpha = 0$ represents our default setting. Increasing α deviates from this, and is expected to gradually weaken the forgetting effect.

Similarly, the retention condition is designed to preserve non-target classes knowledge. It's implemented with $p_i \propto \text{Softmax}(f_{\theta_o}(\mathbf{x})/T)_i$, where $T = 1$ is our default setting. Increasing T deviates from the retention condition and is anticipated to reduce the retention effect.

The experimental results, as shown in Tab. 5, align well with our expectations. When progressively increasing α to violate the forgetting condition, the model's accuracy on $\mathcal{D}_{\text{ft}}$ steadily rises from 0% to 95.00%. Likewise, increasing T to deviate from the retention condition reduces the model's accuracy on $\mathcal{D}_{\text{rt}}$, from 95.03% to 18.81%. It is noteworthy that when varying α to validate the forgetting condition, Acc_{rt} remains stable, while varying T results in the same stability in Acc_{ft}. This confirms the validity of our conditions in effectively controlling forgetting without compromising retention, and vice versa.

5.4. Feature Representation Visualization

To investigate the impact of forgetting on model's feature space, we use t-SNE [52] to visualize the feature distributions. Different colors indicate distinct predicted classes, and triangles are used to mark samples belonging to the for-

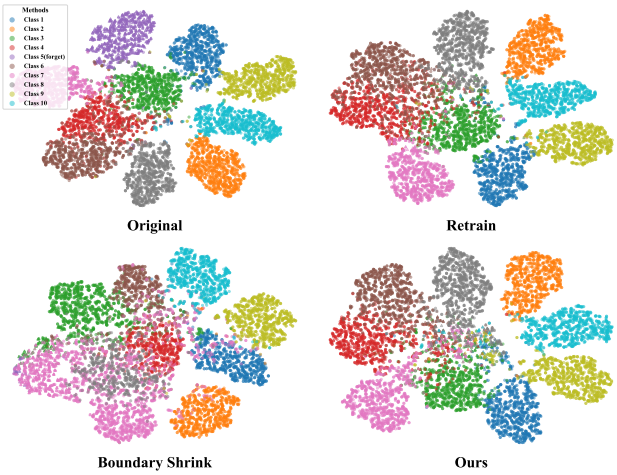
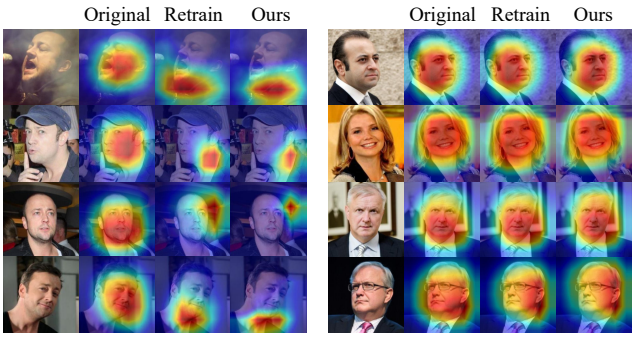


Figure 4. T-SNE visualization of feature representations space.



(a) Forgetting Individuals **(b)** Remaining Individuals

Figure 5. Grad-CAM visualization of face recognition models on forgetting and remaining individuals.

getting class. As illustrated in Fig. 4, the original model exhibits clear class boundaries. Both ours and retrain model successfully unlearn the forget class samples, marked in purple, while preserving the boundaries of the remaining classes. In contrast, Boundary Shrink causes noticeable boundary degradation in certain classes; for example, the pink dots are dispersed across various regions of the space.

6. Application to Downstream Tasks

6.1. Face Recognition with Unlearning

In this part, we apply machine unlearning to face recognition to achieve the objective of safeguarding individual privacy [5, 9]. As shown in the Appendix, our method outperforms all other methods across all metrics and achieves results closest to the retrain model in quantitative comparisons. To further investigate the forgetting effects, we employ Grad-CAM [43] to visualize model attention areas. The heatmaps highlight areas important to each model's predictions. In Fig. 5a, which relates to individuals designated for forgetting, the original model focuses on facial regions, while both the retrain model and our model effectively suppress recognition signals associated with these

(a) Forgetting "potter plant" class **(b)** Forgetting "dog" class

Figure 6. Visualization of unlearning in semantic segmentation.

faces. In Fig. 5b, depicting remaining individuals, all three models maintain strong attention on relevant features, indicating robust knowledge retention for non-forgotten classes.

6.2. Backdoor Defense with Unlearning

Data poisoning is a common backdoor attack method. Employing unlearning methods allows us to eliminate the influence of poisoned data and achieve effective defense. We adopt the recovery algorithm [30] and utilize DELETE to unlearn. The quantitative results in the Appendix show that DELETE achieves improved performance. Notably, for 3×3 size, we observe improvements of 6.96% and 4.19% compared to others. Meanwhile, other methods face challenges of insufficient defense and reduced model performance.

6.3. Semantic Segmentation with Unlearning

We compare the segmented output images of the original model, the retrain model, and ours, as illustrated in Fig. 6. Our model achieves segmentation results similar to those of the retrain model, effectively avoiding the segmentation of the target class. Notably, forgetting a class is not the same as merely mis-segmenting it as background. For example, due to the absence of dog training samples, the retrained model may classify dog as other visually similar categories, such as cat or cow in Fig. 6b. Our approach effectively reproduces this behavior. The quantitative comparison results are provided in the Appendix.

7. Conclusion

In this paper, we introduce a novel machine unlearning method DELETE that requires neither access to remaining data nor intervention during training, aligning closely with real-world application constraints. Leveraging a theoretical framework that decomposes unlearning loss into forgetting and retention terms, our method effectively separates the influence of forgotten and retained classes through a masking distillation approach. Extensive experiments demonstrate its state-of-the-art performance across setups. We also apply our method to downstream tasks, including face recognition, backdoor unlearning, and semantic segmentation, demonstrating its real-world potential.

Acknowledgements

This work was supported partially by NSFC(92470202, U21A20471), National Key Research and Development Program of China (2023YFA1008503), Guangdong NSF Project (No. 2023B1515040025).

References

[1] Alessandro Acquisti. The economics of personal data and the economics of privacy. *Economics*, 2010. 2, 3

[2] Thomas Baumhauer, Pascal Schöttle, and Matthias Zeppelzauer. Machine unlearning: Linear filtration for logit-based classifiers. *Machine Learning*, 2022. 2

[3] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *S&P*, 2021. 1, 2

[4] Jonathan Brophy and Daniel Lowd. Machine unlearning for random forests. In *ICML*, 2021. 2

[5] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *FG*, 2018. 8

[6] Shuo Cao, Yihao Liu, Wenlong Zhang, Yu Qiao, and Chao Dong. Grids: Grouped multiple-degradation restoration with image degradation similarity. In *ECCV*, 2024. 2

[7] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *S&P*, 2015. 2

[8] 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, 2, 3, 5, 6

[9] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *CVPR*, 2023. 1, 3, 4, 5, 8

[10] Yuantao Chen, Jie Xiong, Weihong Xu, and Jingwen Zuo. A novel online incremental and decremental learning algorithm based on variable support vector machine. *Cluster Computing*, 2019. 2

[11] 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. 2, 5

[12] Kate Crawford and Trevor Paglen. Excavating AI: the politics of images in machine learning training sets. *AI Soc.*, 2021. 1

[13] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5

[14] 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. *arXiv preprint arXiv:2310.12508*, 2023. 1, 4, 5

[15] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *AAAI*, 2024. 2

[16] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *ICCV*, 2023. 1

[17] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *CVPR*, 2023. 2

[18] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. In *NeurIPS*, 2019. 2

[19] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *CVPR*, 2020. 1, 2, 3, 5, 6

[20] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *IJCV*, 2021. 2

[21] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI*, 2021. 1, 2

[22] Lucjan Hanzlik, Yang Zhang, Kathrin Grosse, Ahmed Salem, Maximilian Augustin, Michael Backes, and Mario Fritz. Mlcapsule: Guarded offline deployment of machine learning as a service. In *CVPR*, 2021. 2

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[24] Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[25] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *AISTATS*, 2021. 5

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

[27] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *NeurIPS*, 2024. 2, 5

[28] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015. 5

[29] Na Li, Chunyi Zhou, Yansong Gao, Hui Chen, Anmin Fu, Zhi Zhang, and Yu Shui. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *arXiv preprint arXiv:2403.08254*, 2024. 2, 3

[30] Yang Liu, Mingyuan Fan, Cen Chen, Ximeng Liu, Zhuo Ma, Li Wang, and Jianfeng Ma. Backdoor defense with machine unlearning. In *INFOCOM*, 2022. 1, 2, 8

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 5

[32] Zuhao Liu, Xiao-Ming Wu, Dian Zheng, Kun-Yu Lin, and Wei-Shi Zheng. Generating anomalies for video anomaly detection with prompt-based feature mapping. In *CVPR*, 2023. 2

[33] Zhen Lv, Yangqi Long, Congzhentao Huang, Cao Li, Chengfei Lv, Hao Ren, and Dian Zheng. Spatialdreamer: Self-supervised stereo video synthesis from monocular input. *arXiv preprint arXiv:2411.11934*, 2024. 2

[34] Ananth Mahadevan and Michael Mathioudakis. Certifiable machine unlearning for linear models. *arXiv preprint arXiv:2106.15093*, 2021. 2

[35] Qijie Mo, Yipeng Gao, Shenghao Fu, Junkai Yan, Ancong Wu, and Wei-Shi Zheng. Bridge past and future: Overcoming information asymmetry in incremental object detection. In *ECCV*, 2024. 2

[36] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022. 2

[37] Alex Oesterling, Jiaqi Ma, Flavio Calmon, and Himabindu Lakkaraju. Fair machine unlearning: Data removal while mitigating disparities. In *AISTATS*, 2024. 2

[38] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 2

[39] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 2

[40] V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 2

[41] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *CVPR*, 2023. 1

[42] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 1

[43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 8

[44] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. *arXiv preprint arXiv:2302.04222*, 2023. 1

[45] Shaofei Shen, Chenhao Zhang, Yawen Zhao, Alina Bialkowski, Weitong Tony Chen, and Miao Xu. Label-agnostic forgetting: A supervision-free unlearning in deep models. *arXiv preprint arXiv:2404.00506*, 2024. 2

[46] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *S&P*, 2017. 5

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[48] Yu-Ming Tang, Yi-Xing Peng, Jingke Meng, and Wei-Shi Zheng. Rethinking few-shot class-incremental learning: Learning from yourself. In *ECCV*, 2024. 2

[49] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *TNNLS*, 2023. 2

[50] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *EuroS&P*, 2022. 2

[51] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *USENIX Security*, 2022. 3, 4, 5

[52] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 7

[53] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017. 1, 2

[54] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *CVPR*, 2019. 2

[55] Qizhen Weng, Wencong Xiao, Yinghao Yu, Wei Wang, Cheng Wang, Jian He, Yong Li, Liping Zhang, Wei Lin, and Yu Ding. Mlaas in the wild: Workload analysis and scheduling in large-scale heterogeneous gpu clusters. In *NSDI*, 2022. 2

[56] Xiao-Ming Wu, Dian Zheng, Zuhao Liu, and Wei-Shi Zheng. Estimator meets equilibrium perspective: A rectified straight through estimator for binary neural networks training. In *ICCV*, 2023. 2

[57] Xiao-Ming Wu, Jia-Feng Cai, Jian-Jian Jiang, Dian Zheng, Yi-Lin Wei, and Wei-Shi Zheng. An economic framework for 6-dof grasp detection. In *ECCV*, 2024. 2

[58] Yinjun Wu, Edgar Dobriban, and Susan Davidson. Deltagrad: Rapid retraining of machine learning models. In *ICML*, 2020. 2

[59] Guo-Hao Xu, Yi-Lin Wei, Dian Zheng, Xiao-Ming Wu, and Wei-Shi Zheng. Dexterous grasp transformer. In *CVPR*, 2024. 2

[60] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine unlearning: A survey. *ACM Comput. Surv.*, 2023. 2

[61] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *CVPR*, 2022. 1, 2, 3, 4

[62] Hongbo Zhao, Bolin Ni, Junsong Fan, Yuxi Wang, Yuntao Chen, Gaofeng Meng, and Zhaoxiang Zhang. Continual forgetting for pre-trained vision models. In *CVPR*, 2024. 5

[63] Dian Zheng, Xiao-Ming Wu, Shuzhou Yang, Jian Zhang, Jian-Fang Hu, and Wei-Shi Zheng. Selective hourglass mapping for universal image restoration based on diffusion model. In *CVPR*, 2024. 2

[64] Dian Zheng, Xiao-Ming Wu, Zuhao Liu, Jingke Meng, and Wei-shi Zheng. Diffuvolume: Diffusion model for volume based stereo matching. *IJCV*, 2025. 2