LEARNING TO UNLEARN: INSTANCE-WISE UNLEARNING FOR PRE-TRAINED CLASSIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Since the recent advent of regulations for data protection (e.g., the General Data Protection Regulation), there has been increasing demand in deleting information learned from sensitive data in pre-trained models without retraining from scratch. The inherent vulnerability of neural networks towards adversarial attacks and unfairness also calls for a robust method to remove or correct information in an instance-wise fashion, while retaining the predictive performance across remaining data. To this end, we define *instance-wise unlearning*, of which the goal is to delete information on a set of instances from a pre-trained model, by either misclassifying each instance away from its original prediction or relabeling the instance to a different label. We also propose two methods that reduce forgetting on the remaining data: 1) utilizing adversarial examples to overcome forgetting at the representation-level and 2) leveraging weight importance metrics to pinpoint network parameters guilty of propagating unwanted information. Both methods only require the pre-trained model and data instances to forget, allowing painless application to real-life settings where the entire training set is unavailable. Through extensive experimentation on various image classification benchmarks, we show that our approach effectively preserves knowledge of remaining data while unlearning given instances in both single-task and continual unlearning scenarios.

1 INTRODUCTION

Humans remember and forget: efficiently learning useful knowledge yet regulating privately sensitive information and protecting from malicious attacks. Recent advances in large-scale pre-training enable models to memorize massive information for intelligent operations (Radford et al., 2019), but there is a cost. Language models trained on indiscriminately collected data often disclose private information such as occupations, phone numbers, and family background during text generation (Heikkilä, 2022). Vision models trained on numerous image data sometimes misclassify naturally adversarial or adversarially attacked examples with high-confidence (Hendrycks et al., 2021). A naïve solution is to retrain these models from scratch after refining or reweighting their training datasets (Lison et al., 2021; Zemel et al., 2013; Lahoti et al., 2020). However, such post-hoc processing is impractical due to growing volumes of data and substantial cost of large-scale training: while exercising the Right to be Forgotten (Rosen, 2011; Villaronga et al., 2018) may be straightforward to humans, it is not so straightforward in the context of machine learning. This has sparked the field of *machine unlearning*, in which the main goal is to efficiently delete information while preserving information on the remaining data.

While many machine unlearning approaches have shown promising results deleting data from traditional machine learning algorithms (Mahadevan & Mathioudakis, 2021; Ginart et al., 2019; Brophy & Lowd, 2021) as well as DNN-based classifiers (Tarun et al., 2021; Chundawat et al., 2022; Ye et al., 2022; Yoon et al., 2022; Golatkar et al., 2020; Kim & Woo, 2022; Mehta et al., 2022), existing work are built upon assumptions far too restrictive compared to real-life scenarios. First off, many approaches assume a *class-wise* unlearning setup, where the task is to delete information from all data points that belong to a particular class or set of classes. However, data deletion requests are practically received at a per-instance basis, potentially resulting in a set of data points with a mixture of class labels (Heikkilä, 2022; Mehrabi et al., 2021). Another widely used assumption is that at least a subset of the original training data is available at the time of unlearning. In real settings, however, loading the original dataset may not be an option due to data expiration policies or lack of storage for large amounts of data. Lastly, many approaches consider the main objective as removing the previous effect of the deleting data during training. While this is indeed the ideal case, recent work have shown that fulfilling the objective can still lead to information leakage (Suriyakumar & Wilson, 2022), and unlearning mechanisms must explicitly enforce misprediction for tighter security (Graves et al., 2021).

In light of aforementioned limitations, we propose a framework for *instance-wise unlearning* that deletes information with access only to the pre-trained model and the data points requested for unlearning. Instead of undoing the previous influence of deleting data, we pursue a stronger goal where all requested data points are misclassified, preventing collection of information via interpolation of nearby data points. Inspired by work in the continual learning literature (Ebrahimi et al., 2020; Aljundi et al., 2018), we propose two regularization methods that minimize loss in predictive performance on the remaining data, while completely forgetting information on deleting data. Specifically, we 1) generate adversarial examples by attacking each deleting data point with the pre-trained model and retrain on these examples to prevent representation-level forgetting and 2) use weight importance measures from unlearning instances to focus gradient updates more towards parameters responsible for the originally correct classification of such instances. Extensive experiments on CIFAR-10/-100 (Krizhevsky et al., 2009) and ImageNet-1K (Deng et al., 2009) datasets show that our proposed method effectively preserves overall predictive performance, while completely misclassifying images chosen for deletion. Our qualitative analyses also reveal interesting insights, including lack of any discernible pattern in misclassification that may be exploited by adversaries, preservation of the previously learned decision boundary, and forgetting of high-level features within deleted images. In summary, our **main contributions** are as follows:

- We propose *instance-wise unlearning* through intended misclassification, under the assumption that only the pre-trained model and data to forget are available at hand.
- We present two model-agnostic regularization methods that reduce forgetting on remaining data while misclassifying data for deletion.
- Empirical evaluations on well-known image classification benchmarks show that our proposed method significantly boosts predictive performance after unlearning.

2 Related Work

Machine unlearning. Machine unlearning (Cao & Yang, 2015) is a field that makes a pre-trained model forget information learned from a specified subset of data. For this, the existing studies have taken an approach that deletes the influence of unwanted data points (denoted as $\mathcal{D}_f$) from the model while retaining the predictive performance on the rest of the data (denoted as $\mathcal{D}_r$). Mahadevan & Mathioudakis (2021); Ginart et al. (2019); Brophy & Lowd (2021) proposed unlearning methods for a linear/logistic regression, k-means clustering, and random forests, respectively. These methods are specifically designed for simple machine learning models, not for neural networks.

Recently, the machine unlearning for neural networks have been studied in different settings, shown in Table 1. These methods can be categorized into two approaches: *class-wise* and *instance-wise unlearning*. The *class-wise unlearning* is to forget a certain class (*e.g.,* 9-th class of CIFAR-10) while retaining the performance on the remaining class (Tarun et al., 2021; Chundawat et al., 2022; Ye et al., 2022; Yoon et al., 2022). On the other hand, the *instance-wise unlearning* is designed to delete instance-wise information (*e.g.,* several images of CIFAR-10) from the pre-trained model (Golatkar et al., 2020; Kim & Woo, 2022; Mehta et al., 2022). In other words, only instances that are requested to be forgotten should be deleted and the others from the same class should be remembered.

Table 1: Comparison between existing unlearning methods.

Methods	Unit	Goal	D_r	D_f
Tarun et al. (2021)	class	undo	✓	✗
Chundawat et al. (2022)	class	undo	✗	✗
Ye et al. (2022)	class	undo	✗	✓
Yoon et al. (2022)	class	undo	✗	✓
Golatkar et al. (2020)	instance	undo	✓	✓
Kim & Woo (2022)	instance	undo	✓	✓
Mehta et al. (2022)	instance	undo	✓	✓
Our methods	**instance**	**misclassify**	✗	✓

The goal of the existing methods is to make the already trained model identical to the model trained on the dataset with unwanted instances removed (denote as *undo*). Unfortunately, even if the model is trained on the removed dataset, the interpolation capabilities of the neural networks may correctly predict even that we want to erase. This does not lead to complete unlearning in practical applications. Therefore, we define the goal of unlearning as to make the already trained model completely misclassifies the set of instances that should be forgotten (denote as *misclassify*).

Also, the existing methods have different access level to the unlearning data D_f and the rest D_r. The existing solutions for *instance-wise unlearning* require access to the entire dataset (*i.e.,* $\mathcal{D}_r \cup \mathcal{D}_f$). These methods which rely on the availability of the entire data are very far from real-world scenarios. On the other hand, our proposed methods only need to the unlearning dataset (*i.e.,* $\mathcal{D}_f$).

Adversarial examples. Since the vulnerability of neural networks has been revealed (Szegedy et al., 2013), various methods have been proposed to generate adversarial examples that can deceive neural networks (Goodfellow et al., 2014; Kurakin et al., 2016; Madry et al., 2017; Carlini & Wagner, 2017). In the case of white-box attack, an adversarial example can be generated by adding a hardly visible perturbation on a given image based on the gradient information from the model, making the model classify the image to a wrong class. The injected noise of the example is hard to distinguish visually but it causes a serious misclassification of the model. Recently, (Ilyas et al., 2019) experimentally demonstrates that those noise is not meaningless but it rather contains (attack) target label's features for the model.

Weight importance. Weight importance is a measure of how important each weight is when the model predicts an output for a given input data, and it has been used for different purposes, such as weight pruning (Molchanov et al., 2019; Liu et al., 2017; Wen et al., 2016; Alvarez & Salzmann, 2016; Li et al., 2016) and regularization-based continual learning (Kirkpatrick et al., 2017; Aljundi et al., 2018; Chaudhry et al., 2018; Jung et al., 2020; Aljundi et al., 2019). Among them, *regularization-based continual learning* has actively proposed various methods for measuring the weight importance. For overcoming catastrophic forgetting of previous tasks, the weight importance is utilized as the strength of the L2 regularization between a current model's weight and the model's weight trained up to the previous task. Most methods estimate the weight-level importance based on a gradient of a given input data (Kirkpatrick et al., 2017; Aljundi et al., 2018).

3 Method

3.1 Preliminaries and notations

Dataset and pre-trained model. Let $\mathcal{D}_{train}$ be the entire training dataset used to pre-train a classification model $g_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$. We denote $\mathcal{D}_f \subset \mathcal{D}_{train}$ as the unlearning dataset that we want to intentionally forget from the pre-trained model and $\mathcal{D}_r$ as the remaining dataset on which we wish to maintain predictive accuracy ($\mathcal{D}_r := \mathcal{D}_{train} \setminus \mathcal{D}_f$). We denote a pair of an input image $\boldsymbol{x} \in \mathcal{X}$ and its ground-truth label $\boldsymbol{y} \in \mathcal{Y}$ from $\mathcal{D}_{train}$ as $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{D}_{train}$, similarly $(\boldsymbol{x}_f, \boldsymbol{y}_f) \sim \mathcal{D}_f$ and $(\boldsymbol{x}_r, \boldsymbol{y}_r) \sim \mathcal{D}_r$. Also, $\mathcal{D}_{test}$ denotes the test dataset used for evaluation. Note that our approaches assumes access to only the pre-trained model g_θ and the unlearning dataset $\mathcal{D}_f$ during unlearning.

Adversarial examples. The goal of an adversarial attack on an input $(\boldsymbol{x}, \boldsymbol{y})$ is to generate an adversarial example $\boldsymbol{x'}$ that is similar to $\boldsymbol{x}$, but leads to misclassification ($g_{\boldsymbol{\theta}}(\boldsymbol{x'}) \neq \boldsymbol{y}$) when fed to the pre-trained model $g_{\boldsymbol{\theta}}$. In the case of *targeted* adversarial attack, it makes the model predict a specific class different from the true class ($g_{\boldsymbol{\theta}}(\boldsymbol{x'}) = \bar{\boldsymbol{y}}$). The typical optimization form of generating adversarial examples in targeted attack is denoted as

$$\boldsymbol{x'} = \underset{\boldsymbol{z}:\|\boldsymbol{z}-\boldsymbol{x}\|_p \leq \epsilon}{\arg\min} \; \mathcal{L}_{\text{CE}}(g_{\boldsymbol{\theta}}(\boldsymbol{z}), \bar{\boldsymbol{y}}; \boldsymbol{\theta}) \tag{1}$$

where $\mathcal{L}_{\text{CE}}$ stands for the cross-entropy loss. The $\|\boldsymbol{z} - \boldsymbol{x}\|_p \leq \epsilon$ condition requires that the L_p-norm is less than a perturbation budget ϵ. The optimization above is intractable in general, and thus several papers have proposed approximations that can generate adversarial examples without directly solving it (Goodfellow et al., 2014; Kurakin et al., 2016; Carlini & Wagner, 2017; Madry et al., 2017). In this paper, we make use of L_2-PGD targeted attacks Madry et al. (2017) for all experiments.

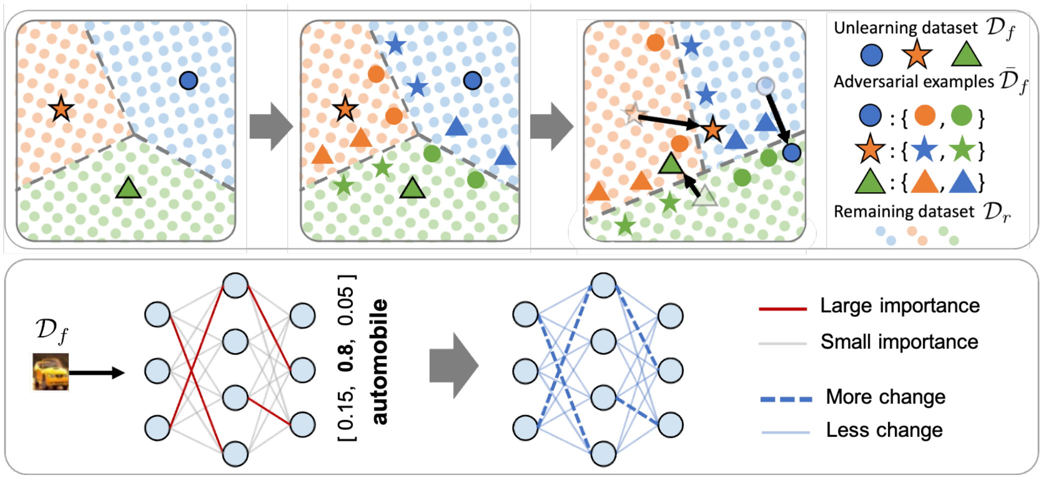


Figure 1: Illustrations of our approaches that reduce forgetting on the remaining data. (Top) Augmenting adversarial examples from unlearning data provides support for preserving the overall decision boundary. (Bottom) Weight importance measures allow us to pinpoint weights we should change to induce misclassification while maintaining other weights to mitigate forgetting.

Measuring weight importance with MAS. To measure weight importance Ω, we consider MAS (Aljundi et al., 2018), an algorithm that estimates weight importance by finding parameters that brings a significant change in the output when perturbed slightly. It estimates the weight importance via a sum of gradients on the L2 norm of the outputs:

$$\Omega_i = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{\partial \|g_{\boldsymbol{\theta}}(\boldsymbol{x}^{(n)};\boldsymbol{\theta})\|_2^2}{\partial \theta_i}\right| \tag{2}$$

where i stands for the index of network parameter weights and $\boldsymbol{x}^{(n)}$ denotes n-th input image from a total of N numbers of images. Note that each Ω_i can be interpreted as a measure of influence or importance of θ_i in producing the output of given N input images.

3.2 Instance-wise unlearning for pre-trained classifiers

Definition of instance-wise unlearning. Let $\hat{g}_{\boldsymbol{\theta}}$ denote the model after unlearning. We consider two types of goals for instance-wise unlearning: **(i)** *misclassifying* all data points in $\mathcal{D}_f$, (*i.e.,* $\hat{g}_{\boldsymbol{\theta}}(\boldsymbol{x}_f) \neq \boldsymbol{y}_f$). **(ii)** *relabeling (or correcting)* the predictions of $\mathcal{D}_f$ (*i.e.,* $\hat{g}_{\boldsymbol{\theta}}(\boldsymbol{x}_f) = \boldsymbol{y}^*_f$) where $\boldsymbol{y}^*_f \neq \boldsymbol{y}_f$ is chosen individually for each input $\boldsymbol{x}_f$. Let $\mathcal{L}_{\text{UL}}$ denote a loss function used for unlearning on a classification model. The above two goals can be realized with the following loss functions:

$$\mathcal{L}^{\text{MS}}_{\text{UL}}(D_f;\boldsymbol{\theta}) = -\mathcal{L}_{\text{CE}}(g_{\boldsymbol{\theta}}(\boldsymbol{x}_f), \boldsymbol{y}_f;\boldsymbol{\theta}) \tag{3}$$

$$\mathcal{L}^{\text{Cor}}_{\text{UL}}(D_f;\boldsymbol{\theta}) = \mathcal{L}_{\text{CE}}(g_{\boldsymbol{\theta}}(\boldsymbol{x}_f), \boldsymbol{y}^*_f;\boldsymbol{\theta}) \tag{4}$$

When unlearning solely based on the two loss functions above, the model is likely to suffer from significant forgetting on $\mathcal{D}_r$. Therefore, a crucial objective shared across both unlearning goals is to overcome forgetting of previously learning knowledge, and maintain as much classification accuracy as possible on $\mathcal{D}_r$.

When both $\mathcal{D}_f$ and $\mathcal{D}_r$ are available, we can easily obtain an *oracle* model that satisfies the objective by re-training the model with the following loss function: $\mathcal{L}_{\text{oracle}}(\mathcal{D}_f, \mathcal{D}_r;\boldsymbol{\theta}) = \mathcal{L}_{\text{UL}}(\mathcal{D}_f;\boldsymbol{\theta}) + \mathcal{L}_{\text{CE}}(\mathcal{D}_r;\boldsymbol{\theta})$. However in real-settings, access to D_r may not be an option due to high cost in data storage. To tackle this limitation, we define regularization-based unlearning for which the goal is to achieve the goal above without explicit use of $\mathcal{D}_r$:

$$\mathcal{L}_{\text{RegUL}}(D_f;\boldsymbol{\theta}) = \mathcal{L}_{\text{UL}}(D_f;\boldsymbol{\theta}) + \mathcal{R}(D_f, g_{\boldsymbol{\theta}}) \tag{5}$$

Here, $\mathcal{R}(\cdot)$ is the regularization term used to overcome forgetting of knowledge on the remaining data $\mathcal{D}_r$. In the following subsections, we introduce two novel regularization methods designed to overcome representation- and weight-level forgetting during the unlearning process.

Algorithm 1 Generate adversarial examples

```
Input: Forgetting data $\mathcal{D}_f$, Model $g_{\boldsymbol{\theta}}$
Output: Adversarial examples $\bar{\mathcal{D}}_r$
$\bar{\mathcal{D}}_r \leftarrow \emptyset$
for $i$ in range $N_f$ do
    $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) \sim \mathcal{D}_f$
    Randomly sample $\bar{\boldsymbol{y}}^{(i)} \neq \boldsymbol{y}^{(i)}$
    for $j$ in range $N_{adv}$ do
        $\boldsymbol{x}'^{(j)}_f \leftarrow L_2\text{-PGD}(\boldsymbol{x}^{(i)}, \bar{\boldsymbol{y}}^{(i)})$ (Eq. 1)
        $\bar{\mathcal{D}}_r \leftarrow \bar{\mathcal{D}}_r \cup \{(\boldsymbol{x}'^{(j)}_f, \bar{\boldsymbol{y}}^{(j)})\}$
    end for
end for
return $\bar{\mathcal{D}}_r$
```

Algorithm 2 Measure weight importance

```
Input: Forgetting data $\mathcal{D}_f$, Model $g_{\boldsymbol{\theta}}$
Output: Weight importance $\bar{\Omega}$
$\bar{\Omega} \leftarrow \{0\}$
$\Omega \leftarrow$ weight importances$(\mathcal{D}_f, g_{\boldsymbol{\theta}})$ (Eq. 2)
for $l$ in range $L$ do
    Get importance of $l$-th layer $\Omega^l \leftarrow \Omega$
    Normalize $\Omega^l \leftarrow \dfrac{\Omega^l - Min(\Omega^l)}{Max(\Omega^l) - Min(\Omega^l)}$
    Update $\bar{\Omega}^l \leftarrow \{1 - \Omega^l\}$
end for
return $\bar{\Omega}$
```

Regularization using adversarial examples. The motivation of using adversarial examples stems from the work of Ilyas et al. (2019), which showed that perturbations added to $\boldsymbol{x}$ to generate an adversarial example $\boldsymbol{x}'$ contain class-specific features of the attack target label $\bar{\boldsymbol{y}} \neq \boldsymbol{y}$. Based on this finding, we utilize generated adversarial examples as part of regularization $\mathcal{R}(\cdot)$ to preserve class-specific knowledge previously learned by the model, overcoming forgetting during unlearning at the *representation-level*. Let D_f be a set of N_f images: $\{(\boldsymbol{x}^{(i)}_f, \boldsymbol{y}^{(i)}_f)\}_{i=1}^{N_f}$. Prior to the unlearning process, we generate adversarial examples $\boldsymbol{x}'_f$ using the targeted PGD attack with a randomly selected attack target label $\bar{\boldsymbol{y}} \neq \boldsymbol{y}_f$. We generate N_{adv} adversarial examples per input $\boldsymbol{x}_f$. Then, we have $\bar{\mathcal{D}}_f = \{(\boldsymbol{x}'^{(k)}_f, \bar{\boldsymbol{y}}^{(k)}_f)\}_{k=1}^{\bar{N}_f}$ where $\bar{N}_f = N_f \times N_{\text{adv}}$. During unlearning, we add $\mathcal{L}_{\text{CE}}(\bar{\mathcal{D}}_f; \boldsymbol{\theta})$ as a regularization term with adversarial examples:

$$\begin{aligned}\mathcal{L}^{\text{Adv}}_{\text{UL}}(D_f; \boldsymbol{\theta}) &= \mathcal{L}_{\text{UL}}(D_f; \boldsymbol{\theta}) + \mathcal{R}_{\text{Adv}}(D_f, g_{\boldsymbol{\theta}}) \\ &= \mathcal{L}_{\text{UL}}(D_f; \boldsymbol{\theta}) + \mathcal{L}_{\text{CE}}(\bar{\mathcal{D}}_f; \boldsymbol{\theta})\end{aligned} \tag{6}$$

An intuitive illustration of this approach in the representation-level is shown in Figure 1. The generated adversarial examples $\bar{\mathcal{D}}_f$ mimic the remaining dataset $\mathcal{D}_r$, providing information of the pre-trained decision boundary within the representation space. As a result, by adding $\mathcal{L}_{\text{CE}}(\bar{\mathcal{D}}_f; \boldsymbol{\theta})$ as a regularizer to the unlearning process, the model can learn a new decision boundary that minimizes $\mathcal{L}_{\text{UL}}$ (in Eq. 3 and 4) while simultaneously attempting to keep the decision boundary of the original model. The pseudocode for generating adversarial examples is in Algorithm 1.

Regularization with weight importance. We also propose a regularization using weight importance to overcome forgetting at the weight-level. As depicted in Figure 1, our approach is to maintain the weights that were less important for $\mathcal{D}_f$ prediction as much as possible, while allowing changes in weights that are considered important for correctly predicting $\mathcal{D}_f$. That is, it is to prevent the weight-level forgetting by penalizing weights that were less important when predicting $\mathcal{D}_f$.

For this, we calculate the weight importance with MAS before unlearning given $g_{\boldsymbol{\theta}}$ and $\mathcal{D}_f$, and normalize the measured importances Ω^l within each l-th layer to lie within $[0, 1]$. Note that this normalized importance Ω^l assigns large values to weights important for $\mathcal{D}_f$. Therefore, we define $\bar{\Omega}^l = 1 - \Omega^l$ as the weight importance for the regularization used for unlearning, so that more important weights are updated more. The objective including weight importance regularization in addition to regularization via adversarial examples can be written as:

$$\begin{aligned}\mathcal{L}^{\text{Adv+Imp}}_{\text{UL}}(D_f; \boldsymbol{\theta}) &= \mathcal{L}^{\text{Adv}}_{\text{UL}}(D_f; \boldsymbol{\theta}) + \mathcal{R}_{\text{Imp}}(D_f, g_{\boldsymbol{\theta}}) \\ &= \mathcal{L}^{\text{Adv}}_{\text{UL}}(D_f; \boldsymbol{\theta}) + \sum_i \bar{\Omega}_i (\theta_i - \tilde{\theta}_i)^2\end{aligned} \tag{7}$$

where i is the index of each weight and $\tilde{\theta}$ is the initial weight of the pre-trained classifier before unlearning. The pseudocode of measuring weight importance is shown in Algorithm 2. Throughout

Table 2: Evaluation results before and after unlearning k instances from ResNet-50 pretrained on respective image classification datasets. While using negative gradients only loses significant information on $\mathcal{D}_r$, our proposed methods ADV and ADV+IMP retain predictive performance on $\mathcal{D}_r$ as well as $\mathcal{D}_{test}$, while completely forgetting instances in $\mathcal{D}_f$.

		CIFAR-10				CIFAR-100				ImageNet-1K			
		$k=4$	$k=16$	$k=64$	$k=128$	$k=4$	$k=16$	$k=64$	$k=128$	$k=4$	$k=16$	$k=64$	$k=128$
D_f (↓)	BEFORE	100.0	100.0	99.38	99.53	100.0	100.0	100.0	100.0	91.66	87.50	84.90	86.72
	NEGGRAD	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	ADV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	ADV+IMP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D_r (↑)	BEFORE	99.60	99.60	99.60	99.60	99.98	99.98	99.98	99.98	87.42	87.42	87.42	87.42
	NEGGRAD	38.44	15.79	9.22	7.11	**99.71**	66.97	26.20	11.64	**83.34**	61.18	40.50	30.16
	ADV	79.40	69.70	66.97	53.49	83.90	89.18	81.07	76.28	74.13	81.09	76.02	69.01
	ADV+IMP	**82.95**	**85.75**	**72.77**	**54.51**	83.89	**89.91**	**89.48**	**82.86**	74.16	**81.77**	**79.36**	**75.33**
D_{test} (↑)	BEFORE	92.59	92.59	92.59	92.59	77.10	77.10	77.10	77.10	76.01	76.01	76.01	76.01
	NEGGRAD	36.56	15.87	9.28	7.11	**74.54**	48.07	21.11	10.19	**72.53**	53.30	35.61	26.73
	ADV	74.34	65.14	62.23	49.47	60.00	63.17	57.43	53.89	62.12	70.42	65.89	59.73
	ADV+IMP	**77.53**	**79.65**	**67.08**	**50.82**	60.50	**63.69**	**62.83**	**58.44**	65.15	**70.97**	**68.72**	**65.09**

various experiments, we observe that applying the regularization using adversarial examples is already effective to overcome the forgetting for knowledge of $\mathcal{D}_r$, and the additional regularization with weight importance further enhances performance even further, especially in more harder scenarios such as continual unlearning. The pseudocode of the overall unlearning pipeline is shown in the supplementary material.

4 EXPERIMENTS

In this section, we evaluate our proposed instance-wise unlearning methods in various image classification benchmarks. We first describe our experimental setup, including datasets, baselines and experimental details. We then show that our methods effectively preserves knowledge of remaining data while unlearning instances that should be forgotten in both single-task and continual unlearning scenarios. Lastly, we offer qualitative analyses on three parts: prediction patterns, decision boundary and layer-wise representations in unlearning.

4.1 SETUP

Datasets and baselines. We evaluate our unlearning methods on three different image classification datasets: CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and ImageNet-1K (Deng et al., 2009). Also, we use the ResNet-50 (He et al., 2016) as a base model. The experimental results of various base models are available in the appendix. The compared methods are as follows: BEFORE, the pre-trained model before unlearning; NEGGRAD (Golatkar et al., 2020), fine-tuning on $\mathcal{D}_f$ using negative gradients (*i.e.* $\mathcal{L}_{\text{UL}}^{\text{MS}}$); CORRECT, fine-tuning using $\mathcal{L}_{\text{UL}}^{\text{Cor}}$; ADV is our proposed method using adversarial examples (*i.e.* $\mathcal{L}_{\text{UL}}^{\text{Adv}}$); ADV+IMP, our unlearning method using both adversarial examples and the weight importance regularization (*i.e.* $\mathcal{L}_{UL}^{Adv+Imp}$).

Experimental details. For each dataset, we randomly pick $k \in \{4, 16, 64, 128\}$ images from the entire training dataset as the unlearning data $\mathcal{D}_f$ and consider the remaining as $\mathcal{D}_r$. For the unlearning, we use a SGD optimizer with a learning rate of 1e-3, weight decay of 1e-5, and momentum of 0.9 across all experiments. We take early stopping when the model attains zero accuracy from the unlearning data $\mathcal{D}_f$. For generating adversarial examples from $\mathcal{D}_f$, we use L_2-PGD targeted attack (Madry et al., 2017) with a learning rate of 1e-1, attack iterations of 100 and $\epsilon = 0.4$. It generates 20 adversarial examples for CIFAR-10 and 200 examples for CIFAR-100 and ImageNet-1K. For the weight importance regularization, we set regularization strength $\lambda = 1$ in Eq. 5.

4.2 MAIN RESULTS

Results on various datasets. Table 2 shows evaluation results before and after unlearning k instances from ResNet-50 models pre-trained on each of three different datasets. With respect to accuracies on $\mathcal{D}_f$, we find that ResNet-50 can completely forget up to $k = 128$ instances with consistently zero post-unlearning accuracies. On CIFAR-10, using negative gradients only results in significant loss of accuracy on the remaining data (i.e. $\mathcal{D}_r$ and $\mathcal{D}_{test}$), performing worse than

Table 3: Results analogous to Table 2, but with unlearning via relabeling each image in $\mathcal{D}_f$ to an arbitrarily chosen class. We see a similar trend where CORRECT loses significant information on $\mathcal{D}_r$, while our proposed methods retain predictive performance on $\mathcal{D}_r$ as well as $\mathcal{D}_{test}$.

		CIFAR-10				CIFAR-100				ImageNet-1K			
		$k = 4$	$k = 16$	$k = 64$	$k = 128$	$k = 4$	$k = 16$	$k = 64$	$k = 128$	$k = 4$	$k = 16$	$k = 64$	$k = 128$
D_f (↑)	BEFORE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	CORRECT	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.84	100.0	100.0	100.0	100.0
	ADV	95.0	100.0	99.375	98.28	90.0	100.0	100.0	98.28	100.0	100.0	87.5	71.32
	ADV+IMP	90.0	100.0	53.75	50.16	80.0	86.25	20.63	15.16	100.0	100.0	8.59	4.30
D_r (↑)	BEFORE	99.60	99.60	99.60	99.60	99.98	99.98	99.98	99.98	87.42	87.42	87.42	87.42
	CORRECT	28.39	11.75	12.33	9.71	**96.14**	74.84	31.79	18.64	**84.34**	82.94	76.21	68.03
	ADV	81.43	85.53	83.36	81.06	69.55	**92.94**	94.64	96.32	70.05	83.09	**84.75**	**84.54**
	ADV+IMP	**83.43**	**91.15**	**94.76**	**90.57**	68.77	90.73	**96.68**	**96.44**	74.18	**83.34**	83.27	80.15
D_{test} (↑)	BEFORE	92.59	92.59	92.59	92.59	77.10	77.10	77.10	77.10	76.01	76.01	76.01	76.01
	CORRECT	27.62	11.79	12.16	9.80	**69.82**	53.11	24.37	14.64	**73.26**	71.90	65.68	58.25
	ADV	76.35	79.15	76.95	74.61	51.23	**65.62**	66.79	68.56	64.81	72.02	**73.41**	**73.32**
	ADV+IMP	**78.08**	**84.24**	**86.92**	**82.82**	50.60	64.28	**69.15**	**68.60**	64.94	**72.20**	71.82	68.92

random-choice when the number of forgetting instances is as large as 128. Meanwhile, adding regularization with adversarial examples boosts the accuracy by more than 40% depending on the number of instances to forget. Incorporating weight importances from MAS provides further improvement. Results from CIFAR-100 and ImageNet-1K show a similar trend except when $k = 4$, where adding our regularization approaches deteriorates performance. This well aligns with our intuition as the model can easily misclassify a small number of examples by tweaking a small number of model parameters, hence forgetting $\mathcal{D}_f$ without losing much information on $\mathcal{D}_r$ and $\mathcal{D}_{test}$ despite lack of regularization. The benefit of using adversarial examples is also small when k is small as the diversity amongst images in $\mathcal{D}_{adv}$ is limited by the number of instances to forget.

Table 3 shows results analogous to Table 2, but with the goal of relabeling data points in $\mathcal{D}_f$ to arbitrarily chosen labels rather than misclassifying. We find that a similar trend, where ADV attains significantly less forgetting in $\mathcal{D}_r$ and $\mathcal{D}_{test}$ compared to CORRECT, while succesfully relabeling all points in most cases. While ADV+IMP show even less forgetting, it loses accuracy in relabeling $\mathcal{D}_f$, showing that regularization via weight importance focuses too much on retaining previous knowledge rather than adapting to corrections provided in $\mathcal{D}_f$. An intuitive explanation on why this occurs particularly in relabeling is that while misclassifying can be done easily by driving the input to its closest decision boundary, relabeling can be difficult if the new class is far from the original class in the representation space. The difficulty rises even more when the size of $\mathcal{D}_f$ is large, in which case more parameters in the network are discouraged from being updated during unlearning.

Correcting natural adversarial examples. Leveraging the ImageNet-A (Hendrycks et al., 2021) dataset consisting of natural images that are misclassified with high-confidence by strong classifiers, we test whether our method can make corrections on these adversarial examples, while preserving knowledge from the original training data. For this experiment, we consider $\mathcal{D}_f$ to consist k adversarial images from ImageNet-A, and adjust a ResNet-50 model pre-trained on ImageNet-1K to correctly classify $\mathcal{D}_f$ via our unlearning framework. Table 4 shows the results for $k = \{16, 32, 64, 128\}$. We find that correcting predictions of a small number of images (e.g. $k = 16$), finetuning the model naïvely with cross-entropy only attains the best accuracy in both $\mathcal{D}_r$ and $\mathcal{D}_{test}$. When correcting larger number of images, however, the absence of regularization terms results in larger forgetting in $\mathcal{D}_r$ compared to ADV and ADV+IMP, with a performance gap that consistently increases with the number of adversarial images. Another takeaway is that regularization via weight importance does not help in this scenario, even showing a significant drop in $\mathcal{D}_f$ accuracy when a large number of adversarial images are introduced. This implies that using weight importances imposes too strong a regularzation that correcting predictions for $\mathcal{D}_f$ itself becomes non-trivial. We conjecture that the aggregation of important parameters for predictions in $\mathcal{D}_f$ cover a large proportion of the network with large k, and that careful search for the Pareto optimal between accuracies on $\mathcal{D}_f$ and on $\mathcal{D}_r$ is required.

Continual unlearning. In real-world scenarios, it is likely that data removal requests come as a stream, rather than all at once. Ultimately, despite continual unlearning requests, we need the unlearning method that can delete the requested data while maintaining performance for the rest data. Thus, we consider the setting of deleting $k = \{8, 16, 64, 128\}$ data by repeating the procedure of continually unlearning $\mathcal{D}_f$ in small fragments of size $k_{CL} = 8$. Table 5 shows the results of continual unlearning in the model trained with ResNet-50 on CIFAR-100. We observe that NEGGRAD suffers

Table 4: Correcting adversarial images from ImageNet-A. Adv achieves the least forgetting, while Adv+Imp fails to correct large number of predictions due to strong regularization.

		ImageNet-A			
		$k=16$	$k=32$	$k=64$	$k=128$
D_f (↑)	Before	0.0	0.0	0.0	0.0
	Correct	100.0	100.0	100.0	100.0
	Adv	100.0	100.0	95.31	83.44
	Adv+Imp	100.0	100.0	10.94	9.38
D_r (↑)	Before	87.46	87.46	87.46	87.46
	Correct	**84.41**	83.29	80.79	77.38
	Adv	81.75	**83.80**	**83.74**	**83.44**
	Adv+Imp	81.82	83.73	83.53	82.86
D_{test} (↑)	Before	76.15	76.15	76.15	76.15
	Correct	**73.21**	72.04	69.91	66.73
	Adv	70.89	**72.58**	**72.68**	**72.36**
	Adv+Imp	70.98	72.51	72.39	71.68

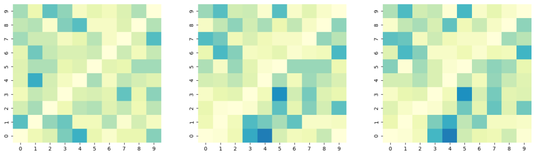
Table 5: Unlearning instances continually by increments of $k_{CL} = 8$ images per step. Our methods outperform NegGrad in the continual unlearning scenario as well.

		CIFAR-100 ($k_{CL} = 8$)			
		$k=8$	$k=16$	$k=64$	$k=128$
D_f (↓)	Before	100.0	100.0	100.0	100.0
	NegGrad	0.0	0.0	0.0	0.52
	Adv	0.0	0.0	1.04	0.0
	Adv+Imp	0.0	0.0	0.0	1.04
D_r (↑)	Before	99.98	99.98	99.98	99.98
	NegGrad	80.58	31.85	6.60	1.89
	Adv	80.33	70.54	59.67	38.16
	Adv+Imp	**81.46**	**72.78**	**62.30**	**47.14**
D_{test} (↑)	Before	77.10	77.10	77.10	77.10
	NegGrad	58.20	24.48	5.73	1.22
	Adv	57.56	50.43	43.48	30.10
	Adv+Imp	**58.33**	**51.97**	**45.09**	**36.17**

from large forgetting as the iteration of unlearning procedure increases. On the other hand, our proposed method shows significantly less forgetting while effectively deleting for $\mathcal{D}_f$ even after multiple iterations of unlearning.

4.3 Qualitative Analysis

Through further analysis, we gather insight on the following questions: **Q1.** Is there any particular pattern in how the model unlearns a set of instances (i.e. does the model use any particular label as a retainer for deleted data)? **Q2.** How does the model isolate out instances in $\mathcal{D}_f$ from its previous decision boundary? **Q3.** How do layer-wise representations of data points in $\mathcal{D}_f$ and $\mathcal{D}_r$ change before and after unlearning? For interpretable visualizations, we perform the following analysis on a ResNet-18 model pre-trained on CIFAR-10.



(a) NegGrad (b) Adv (c) Adv+Imp

Figure 2: Confusion matrices showing average pairwise frequencies of pre- (Y-axis) and post-unlearning (X-axis) prediction labels from $\mathcal{D}_f$. A hue closer to blue indicates higher frequency. Our unlearning framework does not produce any discernible correlation in misclassification.

A1. Our method shows no pattern in misclassification. We first check whether the unlearned model classifies all instances in $\mathcal{D}_f$ to a particular set of labels. The model exhibiting no correlation between true labels and new misclassified labels is crucial with respect to data privacy, as it indicates that the unlearning process avoids the so-called *Streisand effect* where data instances being forgotten unintentionally becomes more noticeable (Golatkar et al., 2020). Figure 2 shows the confusion matrices of (pre-unlearning label, post-unlearning label) pairs from $\mathcal{D}_f$ for $k = 512$. We find no distinguishable pattern when unlearning with our methods as well as NegGrad, which shows that no specific label is used as a retainer, which adds another layer of security against adversaries in search of unlearned data points.

A2. Our method effectively preserves the decision boundary. We check whether the adversarial examples generated from forgetting data help in preserving the decision boundary in the feature space. Figure 3 shows t-SNE (Van der Maaten & Hinton, 2008) visualizations of final-layer activations from examples in $\mathcal{D}_r$ and $\mathcal{D}_f$ before and after unlearning. We find that unlearning through only negative gradient significantly distorts the previous decision boundary, leading to poor predictive performance across $\mathcal{D}_r$. However, when we incorporate adversarial samples from instances in $\mathcal{D}_f$, the decision boundary is well-preserved with unlearned examples being inferred as boundary cases in-between multiple classes. Even for examples that lie far from the decision boundary before unlearning, our method successfully relocates the corresponding representations towards the decision boundary, while keeping each class cluster intact.

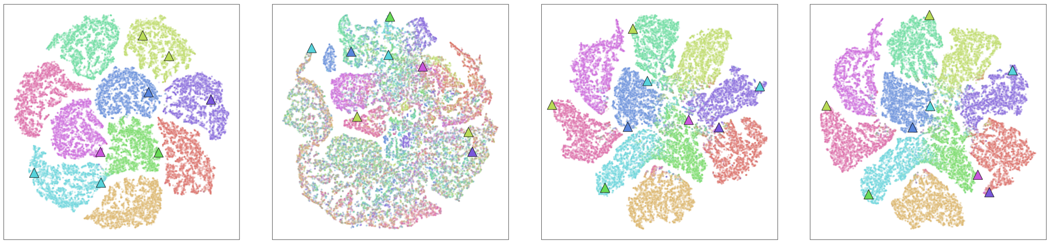

(a) Before (b) NegGrad (c) Adv (d) Adv+Imp

Figure 3: t-SNE plots of CIFAR-10 datapoints in $\mathcal{D}_f$ (triangles) and $\mathcal{D}_r$ (dots) before and after unlearning. Colors indicate true labels for all plots. Regularization with adversarial examples and weight importance effectively preserves the decision boundary while migrating instances in $\mathcal{D}_f$ towards the class boundary to induce misclassification.

A3. Our method unlearns data by forgetting high-level features. Lastly, we compare the representations at each layer of the model before and after unlearning to identify where the intended forgetting occurs. For this analysis, we leverage CKA (Kornblith et al., 2019) which measures correlations between representations given two distinct models. Figure 4 shows the CKA correlation heatmaps between the original ResNet-18 model pre-trained on CIFAR-10 and the same model after unlearning. Results show that for examples in $\mathcal{D}_f$, representations are no longer aligned starting from the 10-th layer while the representations before that layer still resemble those from the original model. This indicates that the model forgets examples by forgetting high-level features, while similarly recognizing low-level features in images as the original model. This insight is consistent with previous observations in the continual learning literature that more forgettable examples exhibit peculiarities in high-level features (Toneva et al., 2018).

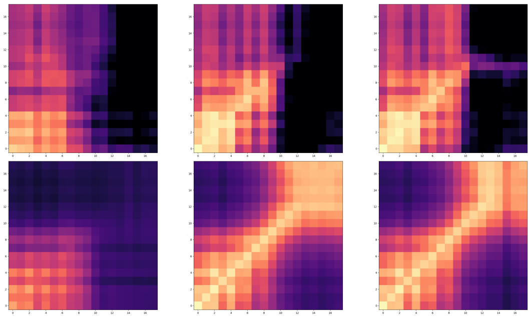

(a) NegGrad (b) Adv (c) Adv+Imp

Figure 4: Layer-wise CKA correlations on $\mathcal{D}_f$ (top row) and $\mathcal{D}_r$ (bottom row) between representations before (X-axis) and after (Y-axis) unlearning. Brighter color indicates higher CKA correlation. NegGrad results in large forgetting of high-level features in not only $\mathcal{D}_f$, but also $\mathcal{D}_r$. Our approaches, on the other hand, selectively forget high-level features only in $\mathcal{D}_f$.

5 Concluding Remarks

We propose an instance-wise unlearning framework that deletes information from a pre-trained model given a set of data instances with mixed labels. Rather than undoing the influence of given instances during the pre-training, we aim for a stronger form of unlearning via intended misclassification. We develop two regularization techniques that reduce forgetting on the remaining data, one utilizing adversarial examples of deleting instances and another leveraging weight importances to focus updates to parameters responsible for propagating information we wish to forget. Both approaches are agnostic to the choice of architecture, and requires access only to the pre-trained model and instances requested for deletion. Experiments on various image classification datasets showed that our methods effectively mitigates forgetting on remaining data, while completely misclassifying deletion data. Further qualitative analyses show that our unlearning framework does not show any pattern in misclassification (i.e. the Streisand effect), preserves the decision boundary with the help of adversarial examples, and unlearns by forgetting high-level features of deleting data. These observations shed light towards future work evaluating the utility our approach as a defense mechanism against membership inference attacks that predict whether a data point was included in the training set by using posterior confidence (Shokri et al., 2017; Salem et al., 2018; Yeom et al., 2018; Sablayrolles et al., 2019) or its distance to nearby decision boundaries (Choquette-Choo et al., 2021; Li & Zhang, 2021). Removing harmful information that lead to socially unfair and biased predictions based upon sensitive traits such as race, gender, and religion (Mehrabi et al., 2021) is another potential contribution from this work.

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

Rahaf Aljundi, Marcus Rohrbach, and Tinne Tuytelaars. Selfless sequential learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Jose M Alvarez and Mathieu Salzmann. Learning the number of neurons in deep networks. *Advances in neural information processing systems*, 29, 2016.

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

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480. IEEE, 2015.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018.

Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pp. 1964–1974. PMLR, 2021.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *arXiv preprint arXiv:2201.05629*, 2022.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *European Conference on Computer Vision*, pp. 386–402. Springer, 2020.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312, 2020.

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

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Melissa Heikkilä. What does gpt-3 ”know” about me?, Aug 2022. URL `https://www.technologyreview.com/2022/08/31/1058800/what-does-gpt-3-know-about-me/`.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271, 2021.

Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Sangwon Jung, Hongjoon Ahn, Sungmin Cha, and Taesup Moon. Continual learning with node-importance based adaptive group sparse regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 3647–3658. Curran Associates, Inc., 2020.

Junyaup Kim and Simon S Woo. Efficient two-stage model retraining for machine unlearning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4361–4369, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019.

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

Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems*, 33:728–740, 2020.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 880–895, 2021.

Pierre Lison, Ildikó Pilán, David Sánchez, Montserrat Batet, and Lilja Øvrelid. Anonymisation models for text data: State of the art, challenges and future directions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4188–4203, 2021.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017.

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

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

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N Ravi. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10422–10431, 2022.

Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Timothy Nguyen, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Architecture matters in continual learning. *arXiv preprint arXiv:2202.00275*, 2022.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11264–11272, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pp. 5558–5567. PMLR, 2019.

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Vinith M. Suriyakumar and Ashia C. Wilson. Algorithms that approximate data removal: New results and limitations, 2022.

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

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *arXiv preprint arXiv:2111.08947*, 2021.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Eduard Fosch Villaronga, Peter Kieseberg, and Tiffany Li. Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Law & Security Review*, 34(2): 304–313, 2018.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Jingwen Ye, Yifang Fu, Jie Song, Xingyi Yang, Songhua Liu, Xin Jin, Mingli Song, and Xinchao Wang. Learning with recoverable forgetting. *arXiv preprint arXiv:2207.08224*, 2022.

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

Youngsik Yoon, Jinhwan Nam, Hyojeong Yun, Dongwoo Kim, and Jungseul Ok. Few-shot unlearning by model inversion. *arXiv preprint arXiv:2205.15567*, 2022.

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

A Appendix

A.1 Pseudo code of overall unlearning process

Algorithm 3 The pseudo code of overall unlearning process the case of using $\mathcal{L}_{\text{UL}}^{\text{MS}}$.

```
UNLEARNACC = 100
MAXEP = 100
EP = 0
$\bar{\mathcal{D}}_r$ ← Generate adversarial examples with Algorithm 1
$\bar{\Omega}$ ← Measure weight importance with Algorithm 2
$\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$
while UNLEARNACC $\neq 0$ do
    Minimize Eqn (6) and (7)
    UNLEARNACC = GetAccuracy($\mathcal{D}_f$, $g_{\boldsymbol{\theta}}$)
    if EP $>$ MAXEP then
        break
        EP += 1
    end if
end while
return $\hat{\boldsymbol{\theta}}$
```

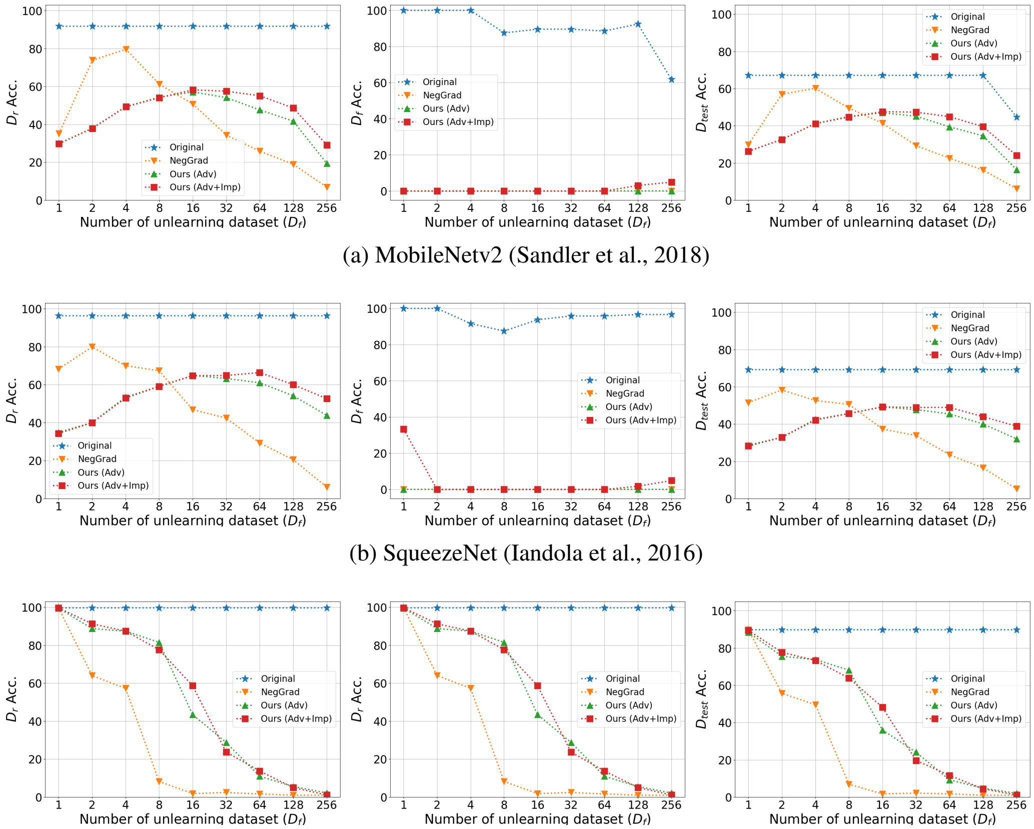
Algorithm 4 The pseudo code of overall unlearning process the case of using $\mathcal{L}_{\text{UL}}^{\text{Cor}}$.

```
UNLEARNACC = 0
MAXEP = 100
EP = 0
$\bar{\mathcal{D}}_r$ ← Generate adversarial examples with Algorithm 1
$\bar{\Omega}$ ← Measure weight importance with Algorithm 2
$\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$
while UNLEARNACC $\neq 100$ do
    Minimize Eqn (6) and (7)
    UNLEARNACC = GetAccuracy($\mathcal{D}_f$, $g_{\boldsymbol{\theta}}$)
    if EP $>$ MAXEP then
        break
        EP += 1
    end if
end while
return $\hat{\boldsymbol{\theta}}$
```

A.2 Additional results on various models

Results on various models. Figure 5 shows unlearning results on CIFAR-100, but with different model architectures. We find that our methods effectively preserve knowledge outside the forgetting data, resulting in up to 40% boost in accuracy. NegGrad again outperforms our methods when $k = 4$, but soon breaks down when unlearning more instances. Interestingly, SqueezeNet and MobileNetv2 suffer from larger forgetting in $\mathcal{D}_r$ and $\mathcal{D}_{test}$ than ResNet-50, possibly due to the width being narrower as previously investigated by Mirzadeh et al. (2022). ViT also suffers from large forgetting, an observation consistent with previous work which showed that ViT suffers more catastrophic forgetting compared to other CNN-based methods in continual learning due to Transformer architectures requiring large amounts of data. We also evaluate the results of unlearning on ImageNet-1K with varying k in Figure 6. Our proposed methods prevent forgetting knowledge about the rest data $\mathcal{D}_r$ better than NegGrad in all cases where k is greater than 8. At the same time, the methods effectively delete information about $\mathcal{D}_f$.

A.3 Supplementary Materials for Rebuttal

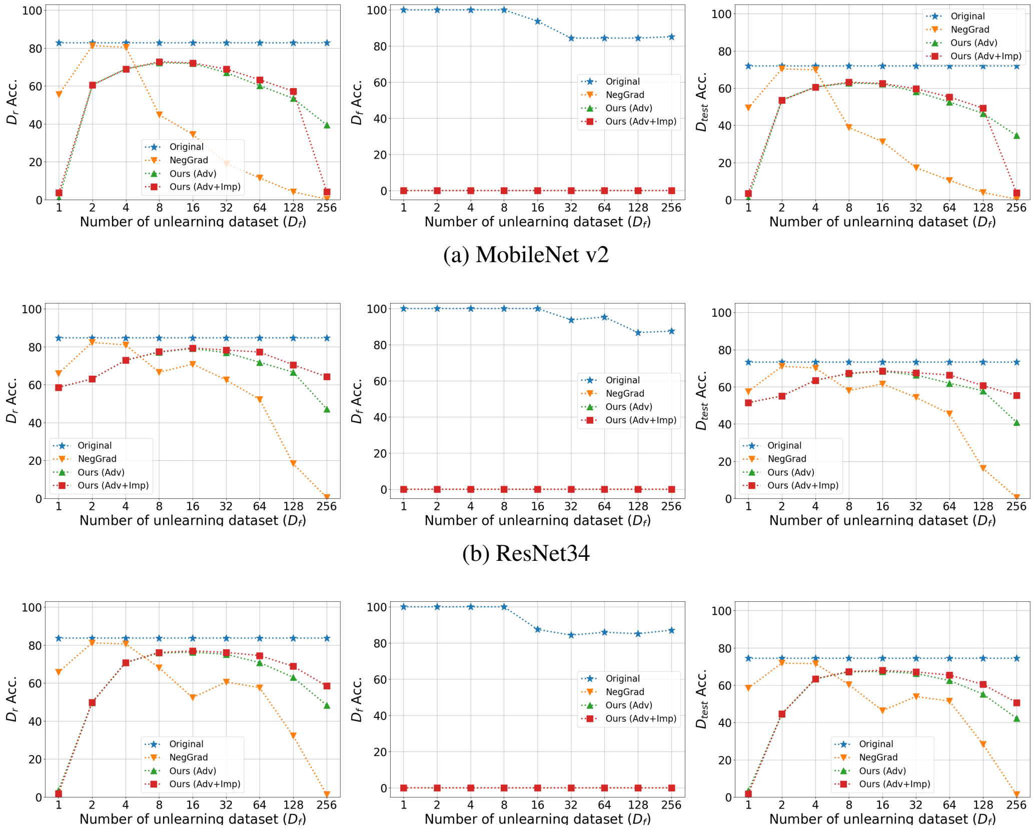


(a) MobileNetv2 (Sandler et al., 2018)

(b) SqueezeNet (Iandola et al., 2016)

(c) ViT (Dosovitskiy et al., 2020)

Figure 5: Experimental results before and after unlearning varying k instances from various models on CIFAR-100.

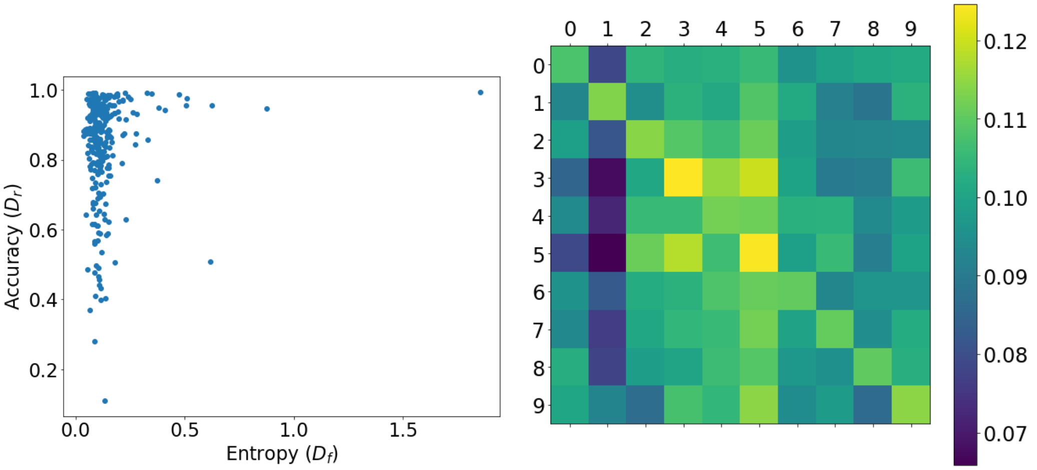


(a) MobileNet v2

(b) ResNet34

(c) DenseNet121

Figure 6: Experimental results before and after unlearning varying k instances from various models on ImageNet-1K.



(a) Analysis for entropy-accuracy

(b) Analysis for a forgotten label

Figure 7: Experimental analysis with CIFAR-10 dataset using ResNet-18. We randomly select single image ($k = 1$) for unlearning and unlearn it with NegGrad. All experiments are conducted with 100 seeds. Each class number denotes a specific label, such as $\{$airplane : 0, automobile : 1, bird : 2, cat : 3, deer : 4, dog : 5, frog : 6, horse : 7, ship : 8, truck : 9$\}$.