Learning to Unlearn: Instance-wise Unlearning for Pre-trained Classifiers

Sungmin Cha[1*], **Sungjun Cho**[2*], **Dasol Hwang**[2*],
Honglak Lee[2], **Taesup Moon**[3] and **Moontae Lee**[2,4]

[1]New York University [2]LG AI Research [3]ASRI / INMC / Seoul National University [4]University of Illinois Chicago
sungmin.cha@nyu.edu, sungjun.cho@lgresearch.ai, dasol.hwang@lgresearch.ai,
honglak@lgresearch.ai, tsmoon@snu.ac.kr, moontae.lee@lgresearch.ai

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 consider *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 pretrained 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.

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 this strength comes at a cost. In computer vision, models trained on numerous image data often misclassify naturally adversarial or adversarially attacked examples with high-confidence (Hendrycks et al. 2021). In natural language processing, models trained on indiscriminately collected data often disclose private information such as occupations, phone numbers, and family background during text generation (Heikkilä 2022).

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, Kieseberg, and Li 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 several machine unlearning approaches have shown promising results deleting data from traditional machine learning algorithms (Mahadevan and Mathioudakis 2021; Ginart et al. 2019; Brophy and 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, Achille, and Soatto 2020; Kim and 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 on 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 (Tarun et al. 2021; Golatkar, Achille, and Soatto 2020; Kim and Woo 2022; Mehta et al. 2022). Unfortunately, loading the original dataset may not be a practical option due to data expiration policies or lack of storage for large data. Lastly, many approaches consider the main objective as *undoing* the previous effect of the deleting data during training. However, previous work have shown that fulfilling this objective can still lead to information leakage (Graves, Nagisetty, and Ganesh 2021).

The aforementioned limitations are also reflected in a recently initialized unlearning challenge (Eleni Triantafillou 2023), centered around a scenario where an age classification model trained on facial images is required to *forget* specific images in order to uphold privacy rights. This marks a departure from previous approaches that primarily adopted a class-wise unlearning setup towards *instance-wise unlearn-*

---

*These authors contributed equally.

ing, while mirroring constraints present in real-world situations such as lack of access to the original training set.

In response, we propose a framework for instance-wise unlearning that deletes information given access only to the pre-trained model and the data points requested for deletion. Instead of undoing the previous influence of deleting data, we pursue a stronger goal where all data points requested for deletion are misclassified, preventing collection of information via interpolation of nearby data points. Achieving this stronger goal for unlearning while maintaining the efficacy of the pre-trained model on the remaining data is a challenging task. Inspired by works in continual learning and adversarial attack literature (Ebrahimi et al. 2020; Aljundi et al. 2018), we thus propose two regularization methods that minimize the loss in predictive performance on the remaining 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, CIFAR-100 (Krizhevsky, Hinton et al. 2009) and ImageNet-1K (Deng et al. 2009) datasets show that our proposed method effectively preserves overall predictive performance, while completely forgetting images requested 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, which can effectively unlearn data with only the pre-trained model and data to forget.

- We present two model-agnostic regularization methods that reduce forgetting on the remaining data while misclassifying data requested for deletion.

- Empirical evaluations on well-known image classification benchmarks show that our proposed method significantly boosts predictive performance after unlearning.

Related Work

Machine unlearning. The main goal of machine unlearning (Cao and Yang 2015) is to adjust a pre-trained model to forget information learned from a specified subset of data. The first set of studies have proposed approaches to delete the influence of unwanted data points from shallow machine learning models such as linear/logistic regression, k-means clustering, and random forests, while retaining the predictive performance on remaining data (Mahadevan and Mathioudakis 2021; Ginart et al. 2019; Brophy and Lowd 2021).

Recent work have studied machine unlearning for deep neural networks in various setups, which we summarize in Table 1. These methods can be categorized into two approaches: *class-wise* and *instance-wise* unlearning. Class-wise unlearning forgets all data points belonging to a certain

Methods	Unit	Goal	$\mathcal{D}_r$	$\mathcal{D}_f$	Updates
Tarun et al. (2021)	class	undo	✓	✗	✗
Graves, Nagisetty, and Ganesh (2021)	class	undo	✗	✗	✓
Chundawat et al. (2022)	class	undo	✗	✗	✗
Ye et al. (2022)	class	undo	✗	✓	✗
Yoon et al. (2022)	class	undo	✗	✓	✗
Golatkar, Achille, and Soatto (2020)	instance	undo	✓	✓	✗
Kim and Woo (2022)	instance	undo	✓	✓	✗
Mehta et al. (2022)	instance	undo	✓	✓	✗
Ours	**instance**	**misclassify**	✗	✓	✗

Table 1: Comparison between existing unlearning methods.

class (*e.g.,* all images of dogs in CIFAR-10) while retaining performance on the remaining classes (Tarun et al. 2021; Chundawat et al. 2022; Ye et al. 2022; Yoon et al. 2022; Graves, Nagisetty, and Ganesh 2021). In contrast, instance-wise unlearning deletes information from individual data points, possibly with mixed classes (Golatkar, Achille, and Soatto 2020; Kim and Woo 2022; Mehta et al. 2022).

Previously, all methods have set the unlearning objective of guiding the pre-trained model towards the model retrained on the dataset after removing unwanted data instances (*i.e.* to *undo* their effect during training). Unfortunately, previous work have shown that this goal fails to provide reliable security against data leakage (Graves, Nagisetty, and Ganesh 2021), due to high interpolation capability of deep neural networks. Therefore, we instead set our goal for unlearning as tuning the pre-trained model towards intentionally *misclassify* the data points requested for deletion. Furthermore, existing methods assume different access levels to the unlearning data $\mathcal{D}_f$ and the remaining data $\mathcal{D}_r$. For our proposed methods, we require access to no more than the unlearning data $\mathcal{D}_f$, allowing seamless application for practitioners under real-world scenarios.

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, Shlens, and Szegedy 2014; Kurakin et al. 2016; Madry et al. 2017; Carlini and Wagner 2017). Interestingly, Ilyas et al. (2019) have experimentally demonstrated that adversarial examples contain useful features about the attack target label for the model. From this observation, we leverage adversarial examples as a means for regularizing the model to preserve its previously learned decision boundary in the feature space.

Weight importance. Weight importance measures how important each parameter is when the model predicts an output for a given input data, and has been used for various purposes such as model pruning (Molchanov et al. 2019; Liu et al. 2017; Wen et al. 2016; Alvarez and 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, Rohrbach, and Tuytelaars 2019; Cha et al. 2021). In the case of regularization-based continual learning, the weight-level 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 for overcoming catastrophic forgetting of previous tasks (Kirkpatrick et al. 2017; Aljundi et al. 2018).
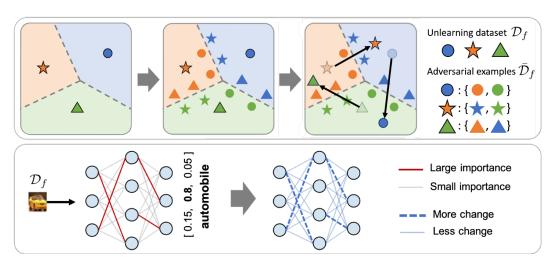
Figure 1: Illustrations of our approaches that reduce forgetting on the remaining data. (Top) Augmenting adversarial examples from unlearning data provides support for preserving the overall decision boundary. (Bottom) Weight importance measures allow us to pinpoint weights we should change to induce misclassification while maintaining other weights to mitigate forgetting.

Method

Preliminaries and notations

Dataset and pre-trained model. Let $\mathcal{D}_{train}$ be the entire training dataset used to pre-train a classification model $g_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$. 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 $g_{\boldsymbol{\theta}}$ 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

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

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 there exist approximations that can generate adversarial examples without directly solving it (Goodfellow, Shlens, and Szegedy 2014; Kurakin et al. 2016; Carlini and Wagner 2017; Madry et al. 2017). For our experiments, we make use of L_2-PGD targeted attacks (Madry et al. 2017).

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 bring 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:

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

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

Our proposed framework

Instance-wise unlearning. Let $\hat{g}_{\boldsymbol{\theta}}$ denote the model after unlearning. We consider two types of misclassification 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 the unlearning loss function used to train a classification model. The two goals above can be realized by minimizing the following loss functions:

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

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

When unlearning solely based on the two loss functions above, the model is likely to suffer from significant *forgetting* on $\mathcal{D}_r$. Thus, a crucial objective shared across both unlearning goals is to prevent forgetting of previously acquired knowledge and classify data in $\mathcal{D}_r$ as accurately as possible.

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 $\mathcal{D}_r$ may not be an option due to

high cost in data storage. To tackle this limitation, we define a regularization-based unlearning that achieves the goal above without use of $\mathcal{D}_r$:

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

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.

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 previously learned class-specific knowledge, overcoming forgetting at the representation-level (Algorithm 1 of the supplementary materials). Let $\mathcal{D}_f$ be a set of N_f images: $\{(\boldsymbol{x}_f^{(i)}, \boldsymbol{y}_f^{(i)})\}_{i=1}^{N_f}$. Prior to unlearning, 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 $\boldsymbol{x}_f$, resulting in a set of $\bar{N}_f = N_f \times N_{\text{adv}}$ examples, denoted as $\bar{\mathcal{D}}_f = \{(\boldsymbol{x}'^{(i)}_f, \bar{\boldsymbol{y}}^{(i)}_f)\}_{i=1}^{\bar{N}_f}$. During unlearning, we add $\mathcal{L}_{\text{CE}}(\bar{\mathcal{D}}_f; \boldsymbol{\theta})$ with adversarial examples as regularization:

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

An intuitive illustration of this approach in the representation-level is shown in Figure 1. In essence, the generated adversarial examples in $\bar{\mathcal{D}}_f$ 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 learns 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.

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 are less important for predictions on $\mathcal{D}_f$ as much as possible, while allowing changes in weights that are considered important for correctly predicting $\mathcal{D}_f$. In other words, we penalize changes of weights that were less important when classifying $\mathcal{D}_f$, thereby preventing weight-level forgetting.

Specifically, we calculate the weight importance with MAS (Aljundi et al. 2018) given $g_{\boldsymbol{\theta}}$ and $\mathcal{D}_f$ before unlearning, and normalize the importance measurements 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 classifying $\mathcal{D}_f$. Therefore, we add a regularization that sums all parameter change in the l-th layer weighted by $\bar{\Omega}^l = 1 - \Omega^l$, such that more important weights are updated more. The objective including weight importance with reg-

ularization via adversarial examples can be written as:

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

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

Experiments

Experimental Setup

Datasets and baselines. We evaluate our unlearning methods on three different image classification datasets: CIFAR-10, CIFAR-100 (Krizhevsky, Hinton et al. 2009), and ImageNet-1K (Deng et al. 2009). We use ResNet-18 as the base model for CIFAR-10, and ResNet-50 for CIFAR-100 and ImageNet-1K (He et al. 2016). Experimental results from other base models such as MobileNetv2 (Sandler et al. 2018), SqueezeNet (Iandola et al. 2016), DenseNet (Huang et al. 2017), and ViT (Dosovitskiy et al. 2020) are also available in the supplementary material.

The compared baselines are as follows: BEFORE corresponds to the pre-trained model before unlearning; ORACLE denotes the ideal model that is retrained with positive gradients from $\mathcal{D}_r$ and negative gradients from $\mathcal{D}_f$ until maximum performance is achieved on $\mathcal{D}_r$ (*i.e.*, $\mathcal{L}_{\text{oracle}}$). In case the accuracy of $\mathcal{D}_f$ does not reach 0 after a certain number of epochs, we report results based on the checkpoint that attained the lowest performance on $\mathcal{D}_f$; NEGGRAD (Golatkar, Achille, and Soatto 2020) fine-tunes the model on $\mathcal{D}_f$ using negative gradients (*i.e.*, $\mathcal{L}_{\text{UL}}^{\text{MS}}$) without any regularization; Repeated adversarial weight perturbation (RAWP) is another baseline derived from AWP (Wu, Xia, and Wang 2020) that induces misclassification by adding weight perturbations generated from $\mathcal{D}_f$ to the model parameters. ADV denotes our proposed method using adversarial examples (*i.e.*, $\mathcal{L}_{\text{UL}}^{\text{Adv}}$); Lastly, ADV+IMP uses adversarial examples as well as weight importance for regularization (*i.e.*, $\mathcal{L}_{\text{UL}}^{\text{Adv+Imp}}$).

Experimental details. For each dataset, we randomly pick $k \in \{16, 64, 128, 256\}$ images from the entire training dataset as unlearning data $\mathcal{D}_f$ and consider the remaining data as $\mathcal{D}_r$. For 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 early-stop training when the model attains 0% or 100% accuracy on the unlearning data $\mathcal{D}_f$, in case of *misclassifying* and *relabeling*, respectively. For generating adversarial examples from $\mathcal{D}_f$, we use L_2-PGD targeted attack (Madry et al. 2017) with a learning

		CIFAR-10				CIFAR-100				ImageNet-1K			
		$k=16$	$k=64$	$k=128$	$k=256$	$k=16$	$k=64$	$k=128$	$k=256$	$k=16$	$k=64$	$k=128$	$k=256$
$\mathcal{D}_f$ (↓)	BEFORE	100.0	99.38	99.53	99.38	100.0	100.0	100.0	100.0	87.50	84.69	86.09	86.02
	ORACLE	0.0	0.0	0.0	0.63	0.0	0.0	0.0	0.0	0.0	0.0	0.26	5.73
	NEGGRAD	0.0	0.0	0.0	3.83	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	RAWP	0.0	0.0	4.06	6.17	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	2.73
$\mathcal{D}_r$ (↑)	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
	ORACLE	98.74	99.72	98.97	39.90	99.96	96.17	96.74	96.43	73.36	77.81	67.7	61.61
	NEGGRAD	15.79	9.22	7.11	6.34	66.97	26.20	11.64	1.70	65.02	35.66	19.27	0.48
	RAWP	67.15	37.32	13.28	9.12	72.44	17.39	2.82	1.12	20.33	2.72	0.29	0.10
	ADV	69.70	66.97	53.49	39.33	89.18	81.07	76.28	65.67	80.92	73.60	66.31	43.70
	ADV+IMP	**85.75**	**72.77**	**54.51**	**48.95**	**89.91**	**89.48**	**82.86**	**67.60**	**81.43**	**79.03**	**74.14**	**65.32**
$\mathcal{D}_{test}$ (↑)	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
	ORACLE	90.21	91.01	89.44	38.59	64.41	67.06	66.88	65.31	63.91	68.08	59.09	54.03
	NEGGRAD	15.87	9.28	7.11	6.47	48.07	21.11	10.19	1.71	56.70	31.10	17.08	0.46
	RAWP	63.10	35.47	12.91	9.12	50.99	14.41	2.79	1.12	17.53	2.41	0.28	0.10
	ADV	65.14	62.23	49.47	36.69	63.17	57.43	53.89	46.45	70.36	65.51	57.34	38.25
	ADV+IMP	**79.65**	**67.08**	**50.82**	**45.44**	**63.69**	**62.83**	**58.44**	**48.71**	**70.77**	**68.43**	**64.05**	**56.52**

Table 2: Evaluation results before and after unlearning k instances from ResNet-18 (for CIFAR-10) and ResNet-50 (for CIFAR-100 and ImageNet-1K) 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 $\mathcal{D}_f$.

rate of 1e-1, attack iterations of 100 and $\epsilon = 0.4$. It generates 20 adversarial examples per image for CIFAR-10 and 200 examples per image for CIFAR-100 and ImageNet-1K. For the weight importance regularization, we set regularization strength $\lambda = 1$ in Eq. 5.

Main Results

Results on various datasets. Table 2 shows results before and after unlearning (misclassifying) k instances from ResNet-18 and ResNet-50 models pre-trained on CIFAR-10 and CIFAR-100/ImageNet-1K, respectively. In terms of accuracies on $\mathcal{D}_f$, we find that all methods can almost forget up to $k = 256$ instances with consistently zero post-unlearning accuracy, except for some cases (*e.g.*, ORACLE, NEGGRAD and RAWP in CIFAR-10 when $k = 256$). For most cases, ORACLE shows not only superior accuracy in the remaining data (i.e. $\mathcal{D}_r$ and $\mathcal{D}_{test}$) but also almost zero accuracy in $\mathcal{D}_f$. Nevertheless, we find that as k increases, its performance decreases, demonstrating the difficulty of achieving the proposed goal of unlearning. NEGGRAD results in a significant loss of accuracy on the remaining data for all datasets. RAWP achieves competitive accuracy in the remaining data than NEGGRAD in CIFAR-10. However, it performs significantly worse when the dataset becomes complex (*e.g.,* CIFAR-100 and ImageNet) or when k becomes larger than 16, showing that simply perturbing the parameters cannot be a direct solution.

Meanwhile, adding regularization with adversarial examples (ADV) boosts the accuracy by more than 40% depending on the number of instances to forget. Incorporating weight importances from MAS (ADV+IMP) provides further improvement, even surpassing the performance of ORACLE in some cases (*e.g.*, $k = 256$ in CIFAR-10 and ImageNet).

For comparison with existing unlearning methods, we also conduct experiments with AMNESIAC (Graves, Nagisetty, and Ganesh 2021) as a baseline, details on which

		UTKFace				
		$k=1$	$k=2$	$k=4$	$k=8$	$k=16$
$\mathcal{D}_f$ (↓)	BEFORE	100.00	100.00	100.00	100.00	100.00
	NEGGRAD	0.00	0.00	0.00	0.00	0.00
	ADV	0.00	0.00	0.00	0.00	0.00
	ADV+IMP	0.00	0.00	0.00	0.00	0.00
$\mathcal{D}_r$ (↑)	BEFORE	99.85	99.85	99.84	99.84	99.83
	NEGGRAD	75.75	36.80	28.95	28.35	32.75
	ADV	79.20	62.69	80.63	72.94	56.17
	ADV+IMP	**87.07**	**63.85**	**82.30**	**74.33**	**57.21**
$\mathcal{D}_{test}$ (↑)	BEFORE	87.14	87.14	87.14	87.14	87.14
	NEGGRAD	72.67	36.57	29.40	27.16	32.14
	ADV	76.30	58.46	78.28	65.50	53.65
	ADV+IMP	**82.24**	**61.79**	**79.54**	**66.98**	**54.37**

Table 3: Instance-wise unlearning results from a ResNet-18 age classifier trained on UTKFace.

can be found in our supplementary material. When the data to be forgotten is small under class-wise unlearning, AMNESIAC is effective without significantly changing the rest of the model. However, when attempting to remove multiple data points with mixed classes, its performance degrades significantly unless retrained on $\mathcal{D}_r$. Unlike our realistic unlearning setup, AMNESIAC requires a record of parameter updates from training on sensitive data. This is viable only when sensitive data is known prior to model pre-training, which makes the method inapplicable in our framework.

Lastly, we mirror a real-world setting and empirically test our instance-wise unlearning framework on facial image data. Specifically, we first pretrain ResNet-18 on age-group classification on the UTKFace dataset consisted of 20k facial images, each labeled with the age of the subject ranging between 0 and 116 (Zhang, Song, and Qi 2017). Following previous work, we divide the age range into three ($[0, 19]$, $[20, 40]$, and $[40, 116]$), effectively reducing the task into 3-way classification (Jung et al. 2021). Ta-

		CIFAR-100 ($k_{CL} = 8$)			
		$k = 8$	$k = 16$	$k = 64$	$k = 128$
$\mathcal{D}_f$ ($\downarrow$)	BEFORE	100.0	100.0	100.0	100.0
	NEGGRAD	0.0	0.0	0.0	1.08
	ADV	0.0	0.0	0.0	0.0
	ADV+IMP	0.0	0.0	0.0	0.56
$\mathcal{D}_r$ ($\uparrow$)	BEFORE	99.98	99.98	99.98	99.98
	NEGGRAD	75.68	30.40	7.00	1.51
	ADV	87.02	71.84	60.12	39.30
	ADV+IMP	**87.42**	**72.99**	**61.76**	**48.04**
$\mathcal{D}_{test}$ ($\uparrow$)	BEFORE	77.10	77.10	77.10	77.10
	NEGGRAD	54.51	24.76	6.22	0.98
	ADV	62.76	51.20	43.40	31.11
	ADV+IMP	**63.01**	**52.98**	**45.55**	**36.87**

Table 4: Unlearning instances continually by increments of $k_{CL} = 8$ images per step. Our methods outperform NEG-GRAD in the continual unlearning scenario as well.

		ImageNet-A			
		$k = 16$	$k = 32$	$k = 64$	$k = 128$
$\mathcal{D}_f$ ($\uparrow$)	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
$\mathcal{D}_r$ ($\uparrow$)	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
$\mathcal{D}_{test}$ ($\uparrow$)	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: Correcting adversarial images in ImageNet-A. ADV achieves the least forgetting, while ADV+IMP fails to correct large number of predictions due to strong regularization.

ble 9 shows the results. For NEGGRAD, we find that its performance degrades quickly, with its classification performance being lower than that of a random baseline (*i.e.*, 33%) when unlearning more than 4 instances. On the other hand, both ADV and ADV+IMP consistently outperform the naïve baseline across all k, showing that our regularization techniques are also effective in the facial image domain. The gap between ADV and ADV+IMP is much larger with small k, which demonstrates high potential of weight importance-based regularization for unlearning under a continuous stream of instance-wise data deletion requests.

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 4 shows the results of continual unlearning in the model trained with ResNet-50 on CIFAR-100. We observe that NEGGRAD suffers 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.

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. For this setup, we employ a baseline CORRECT that simply finetunes the model on $\mathcal{D}_f$ via cross-entropy loss (*i.e.*, $\mathcal{L}_{\text{UL}}^{\text{Cor}}$). In Table 5, we find that naïve finetuning attains the best accuracy in both $\mathcal{D}_r$ and $\mathcal{D}_{test}$ when correcting predictions of a small number of images (e.g. $k = 16$).

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 regularization 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.
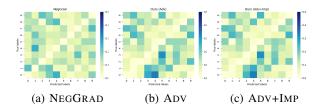

(a) NEGGRAD   (b) ADV   (c) ADV+IMP

Figure 2: Confusion matrices from CIFAR-10 showing average pairwise frequencies of pre- (Y-axis) and post-unlearning (X-axis) prediction labels from $\mathcal{D}_f$. A blue color indicates higher frequency. Our unlearning framework does not produce any discernible correlation in misclassification.

Qualitative Analysis

With 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 due to unlearning? For interpretable visualization, we perform the following analysis on a ResNet-18 model pre-trained on CIFAR-10.
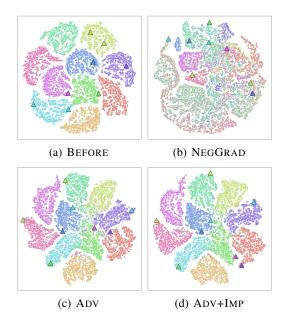
(a) BEFORE   (b) NEGGRAD

(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.

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 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, Achille, and Soatto 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 $\mathcal{D}_f$ help in preserving the decision boundary in the feature space. Figure 3 shows t-SNE (Van der Maaten and 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.
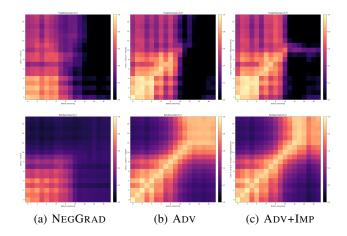

(a) NEGGRAD   (b) ADV   (c) ADV+IMP

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 both $\mathcal{D}_f$ and $\mathcal{D}_r$. Our approaches selectively forget high-level features only in $\mathcal{D}_f$.

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) that measures correlations between representations given two distinct models. Figure 4 shows the CKA correlation heatmaps between the original pre-trained model and the model after unlearning. Results show that for examples in $\mathcal{D}_f$, representations are no longer aligned starting from the 10-th layer, while 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).

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 goal of unlearning via misclassifying or relabeling. We develop two regularization techniques that reduce forgetting on remaining data, one utilizing adversarial examples of deleting instances and another leveraging weight importance to focus updates on parameters responsible for propagating information we wish to forget. Both approaches are agnostic to the choice of architecture and require access only to the pre-trained model and instances requested for deletion. Experiments on various image classification datasets showed that our methods effectively mitigate forgetting on remaining data, while completely misclassifying the deleting data.

Acknowledgments

Pseudocode of our unlearning pipelines

In this section, we provide pseudocode of our instance-wise unlearning framework. Algorithms 3 and 4 show the details of two frameworks, one for unlearning via misclassification and another for unlearning via relabeling. The two pipelines are identical, except for the condition at which unlearning ends: when all examples in $\mathcal{D}_f$ are misclassified vs. when all examples in $\mathcal{D}_f$ are classified correctly to their new labels.

Algorithm 1: Generate adversarial examples

Require: Forgetting data $\mathcal{D}_f$, Model $g_{\boldsymbol{\theta}}$
Ensure: Adversarial examples $\bar{\mathcal{D}}_f$
 1: $\bar{\mathcal{D}}_f \leftarrow \emptyset$
 2: **for** i **in range** N_f **do**
 3:   $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) \sim \mathcal{D}_f$
 4:   Randomly sample $\bar{\boldsymbol{y}}^{(i)} \neq \boldsymbol{y}^{(i)}$
 5:   **for** j **in range** N_{adv} **do**
 6:     $\boldsymbol{x}_f'^{(j)} \leftarrow L_2\text{-PGD}(\boldsymbol{x}^{(i)}, \bar{\boldsymbol{y}}^{(i)})$ (Eq. 1)
 7:     $\bar{\mathcal{D}}_f \leftarrow \bar{\mathcal{D}}_f \cup \{(\boldsymbol{x}_f'^{(j)}, \bar{\boldsymbol{y}}^{(j)})\}$
 8:   **end for**
 9: **end for**
10: **return** $\bar{\mathcal{D}}_f$

Algorithm 2: Measure weight importance

Require: Forgetting data $\mathcal{D}_f$, Model $g_{\boldsymbol{\theta}}$
Ensure: Weight importance $\bar{\Omega}$
 1: $\bar{\Omega} \leftarrow \{0\}$
 2: $\Omega \leftarrow \text{weight importances}(\mathcal{D}_f, g_{\boldsymbol{\theta}})$ (Eq. 2)
 3: **for** l **in range** L **do**
 4:   Get importance of l-th layer $\Omega^l \leftarrow \Omega$
 5:   Normalize $\Omega^l \leftarrow \dfrac{\Omega^l - Min(\Omega^l)}{Max(\Omega^l) - Min(\Omega^l)}$
 6:   Update $\bar{\Omega}^l \leftarrow \{1 - \Omega^l\}$
 7: **end for**
 8: **return** $\bar{\Omega}$

Additional experimental results

Results on relabeling. Table 6 shows results analogous to Table 1 of the manuscript, 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, first, ORACLE achieves superior accuracy in most cases but we observe that it suffers from the trade-off between D_f and the remaining datasets when the size of k become large

---

Algorithm 3: Unlearning via misclassification (i.e., $\mathcal{L}_{\text{UL}}^{\text{MS}}$).

 1: UNLEARNACC = 100
 2: MAXEP = 100
 3: EP = 0
 4: $\bar{\mathcal{D}}_r \leftarrow$ Generate adversarial examples with Algorithm 1
 5: $\bar{\Omega} \leftarrow$ Measure weight importance with Algorithm 2
 6: $\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$ // Set the pretrained model for unlearning
 7: **while** UNLEARNACC $\neq 0$ **do**
 8:   Minimize Eqn (6) or (7)
 9:   UNLEARNACC = GetAccuracy($\mathcal{D}_f, g_{\tilde{\boldsymbol{\theta}}}$)
10:   **if** EP > MAXEP **then**
11:     break
12:     EP += 1
13:   **end if**
14: **end while**
15: **return** $\tilde{\boldsymbol{\theta}}$

Algorithm 4: Unlearning via relabeling (i.e., $\mathcal{L}_{\text{UL}}^{\text{Cor}}$).

 1: UNLEARNACC = 0
 2: MAXEP = 100
 3: EP = 0
 4: $\bar{\mathcal{D}}_r \leftarrow$ Generate adversarial examples with Algorithm 1
 5: $\bar{\Omega} \leftarrow$ Measure weight importance with Algorithm 2
 6: $\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$ // Set the pretrained model for unlearning
 7: **while** UNLEARNACC $\neq 100$ **do**
 8:   Minimize Eqn (6) or (7)
 9:   UNLEARNACC = GetAccuracy($\mathcal{D}_f, g_{\tilde{\boldsymbol{\theta}}}$)
10:   **if** EP > MAXEP **then**
11:     break
12:     EP += 1
13:   **end if**
14: **end while**
15: **return** $\tilde{\boldsymbol{\theta}}$

Algorithm 5: Unlearning via misclassification with RAWP.

 1: UNLEARNACC = 100
 2: MAXEP = 100
 3: EP = 0
 4: $\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$ // Set the pretrained model for unlearning
 5: $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$ // Set the pretrained model for AWP
 6: **while** UNLEARNACC $\neq 0$ **do**
 7:   $\hat{\boldsymbol{\theta}}_p \leftarrow \hat{\boldsymbol{\theta}}$ // Set a proxy model
 8:   Minimize $\mathcal{L}_{\text{UL}}^{\text{MS}}$ for $\hat{\boldsymbol{\theta}}_p$
 9:   Get gradients and update $\hat{\boldsymbol{\theta}}_p$
10:   Get difference $\hat{\boldsymbol{\theta}}_d$ of each layer i following
    : $\hat{\boldsymbol{\theta}}_{d,i} = ||\tilde{\boldsymbol{\theta}}_i||_2 / (||\tilde{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}_{p,i}||_2 + \epsilon) * (\tilde{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}_{p,i})$
11:   Perturb $\tilde{\boldsymbol{\theta}}$ of each layer i following
    : $\tilde{\boldsymbol{\theta}}_i = \tilde{\boldsymbol{\theta}}_i + \gamma \hat{\boldsymbol{\theta}}_{d,i}$
12:   UNLEARNACC = GetAccuracy($\mathcal{D}_f, g_{\tilde{\boldsymbol{\theta}}}$)
13:   **if** EP > MAXEP **then**
14:     break
15:     EP += 1
16:   **end if**
17: **end while**
18: **return** $\hat{\boldsymbol{\theta}}$

Table 6: Results analogous to Table 1 of the manuscript, 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 | | | |
		$k=16$	$k=64$	$k=128$	$k=256$	$k=16$	$k=64$	$k=128$	$k=256$	$k=16$	$k=64$	$k=128$	$k=256$
	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
	ORACLE	100.0	100.0	100.0	77.34	100.0	100.0	100.0	79.92	100.0	100.0	99.48	95.83
$\mathcal{D}_f$ (↑)	CORRECT	100.0	100.0	100.0	100.0	100.0	100.0	99.84	99.84	100.0	100.0	100.0	100.0
	RAWP	100.0	100.0	100.0	96.41	100.0	99.38	99.06	93.75	100.0	98.96	56.25	39.06
	ADV	100.0	99.38	98.28	96.17	100.0	100.0	98.28	92.42	100.0	89.06	71.88	58.59
	ADV+IMP	100.0	53.75	50.16	59.06	86.25	20.63	15.16	9.69	100.0	9.38	3.91	2.86
	BEFORE	99.60	99.60	99.60	99.60	99.98	99.98	99.98	99.98	87.48	87.42	87.42	87.42
	ORACLE	99.73	99.94	99.90	95.81	99.90	99.97	99.79	99.50	83.93	83.18	84.97	83.61
$\mathcal{D}_r$ (↑)	CORRECT	11.75	12.33	9.71	8.97	74.84	31.79	18.64	6.93	82.47	77.21	68.19	44.47
	RAWP	79.94	63.95	52.12	21.14	59.36	40.93	9.20	6.15	48.66	31.55	0.46	0.24
	ADV	85.53	83.36	81.06	72.41	**92.94**	94.64	96.32	**94.28**	83.39	**84.79**	**84.52**	**84.11**
	ADV+IMP	**91.15**	**94.76**	**90.57**	**81.82**	90.73	**96.68**	**96.44**	92.40	**83.58**	83.31	80.43	79.29
	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
	ORACLE	91.65	91.99	91.57	87.51	74.05	74.93	74.15	73.81	72.76	72.04	73.83	72.79
$\mathcal{D}_{test}$ (↑)	CORRECT	11.79	12.16	9.80	8.87	53.11	24.37	14.64	5.95	71.57	66.61	58.37	38.29
	RAWP	73.58	59.48	48.02	20.25	41.93	29.51	7.88	5.71	42.40	27.52	0.48	0.23
	ADV	79.15	76.95	74.61	66.80	**65.62**	66.79	68.56	**66.75**	72.34	**73.51**	**73.38**	**73.00**
	ADV+IMP	**84.24**	**86.92**	**82.82**	**74.68**	64.28	**69.15**	**68.60**	64.81	**72.49**	71.92	69.20	68.46

Table 7: Experimental results for various γ of RAWP.

| | | CIFAR-10 | | | |
		$k=16$	$k=64$	$k=128$	$k=256$
RAWP ($\gamma=0.001$)	$\mathcal{D}_f$	0.0	0.0	0.0	0.0
	$\mathcal{D}_r$	88.31	65.44	42.10	14.81
	$\mathcal{D}_{test}$	81.66	60.90	39.16	14.34
RAWP ($\gamma=0.005$)	$\mathcal{D}_f$	0.0	0.0	0.0	2.89
	$\mathcal{D}_r$	81.70	55.09	28.60	13.45
	$\mathcal{D}_{test}$	75.80	51.79	27.21	13.11
RAWP ($\gamma=0.01$)	$\mathcal{D}_f$	0.0	0.0	4.06	6.17
	$\mathcal{D}_r$	67.15	37.33	13.28	9.12
	$\mathcal{D}_{test}$	63.10	35.47	12.91	9.12
RAWP ($\gamma=0.1$)	$\mathcal{D}_f$	0.9	6.25	8.43	9.61
	$\mathcal{D}_r$	10.97	9.16	10.00	10.00
	$\mathcal{D}_{test}$	11.13	9.09	10.00	10.00

Table 8: Comparison against AMNESIAC unlearning (Graves, Nagisetty, and Ganesh 2021).

| | | CIFAR-100 | | | |
		$k=16$	$k=64$	$k=128$	$k=256$
	BEFORE	95.83	99.83	99.83	99.84
	NEGGRAD	0.0	0.0	0.0	0.0
$\mathcal{D}_f$ (↓)	AMNESIAC	*64.58*	*65.10*	*48.44*	*11.46*
	ADV	0.0	0.0	0.0	0.0
	ADV+IMP	0.0	0.0	6.25	13.02
	BEFORE	99.82	99.48	99.74	99.61
	NEGGRAD	87.88	50.85	32.92	9.66
$\mathcal{D}_r$ (↑)	AMNESIAC	99.07	74.72	51.09	15.07
	ADV	89.14	75.10	65.91	35.57
	ADV+IMP	92.31	79.50	68.34	38.12
	BEFORE	83.88	83.88	83.88	83.88
	NEGGRAD	64.74	39.97	26.78	9.47
$\mathcal{D}_{test}$ (↑)	AMNESIAC	79.99	39.63	42.75	39.05
	ADV	65.95	55.78	46.74	25.15
	ADV+IMP	70.52	60.88	50.34	29.90

(*e.g.*, $k=256$). Second, RAWP shows much better performance than CORRECT in CIFAR-10, however, not for other datasets. Third, ADV attains significantly less forgetting in $\mathcal{D}_r$ and $\mathcal{D}_{test}$ compared to CORRECT, while succesfully relabeling all points in most cases. On the other hand, 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.

Results with other architectures. Figure **??** 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 **??**. Our proposed

		UTKFace				
		$k=1$	$k=2$	$k=4$	$k=8$	$k=16$
D_f ($\downarrow$)	BEFORE	100.00	100.00	100.00	100.00	100.00
	NEGGRAD	0.00	0.00	0.00	0.00	0.00
	ADV	0.00	0.00	0.00	0.00	0.00
	ADV+IMP	0.00	0.00	0.00	0.00	0.00
D_r ($\uparrow$)	BEFORE	99.85	99.85	99.84	99.84	99.83
	NEGGRAD	75.75	36.80	28.95	28.35	32.75
	ADV	79.20	62.69	80.63	72.94	56.17
	ADV+IMP	**87.07**	**63.85**	**82.30**	**74.33**	**57.21**
D_{test} ($\uparrow$)	BEFORE	87.14	87.14	87.14	87.14	87.14
	NEGGRAD	72.67	36.57	29.40	27.16	32.14
	ADV	76.30	58.46	78.28	65.50	53.65
	ADV+IMP	**82.24**	**61.79**	**79.54**	**66.98**	**54.37**

Table 9: Instance-wise unlearning results from a ResNet-18 age classifier trained on UTKFace.

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 $\mathcal{D}_f$.

Comparison against AMNESIAC unlearning. Table 8 shows the results before and after unlearning k instances. To compare AMNESIAC (Graves, Nagisetty, and Ganesh 2021) with our methods, we use ResNet-18 models pretrained on CIFAR-100. Note that these models are sourced from the official code provided by (Graves, Nagisetty, and Ganesh 2021). AMNESIAC is known that an effective method for unlearning a small number of k, without loss of performance through the removal of parameter updates for sensitive data. However, as we already shown in Table 1 of the manuscript, it's important to mention that (Graves, Nagisetty, and Ganesh 2021) doesn't employ both $\mathcal{D}_f$ and $\mathcal{D}_r$ for class-wise unlearning. Instead, they necessitate the preservation of model parameter updates from each batch containing sensitive data. However, this requirement might not be feasible for real-world application due to its impracticality. Also, it has been proven that it is effective for *class-wise unlearning*.

Aligned with the manuscript, we focus to demonstrate the efficiency of both our proposed method and AMNESIAC in more difficult *instance-wise unlearning*. Overall, the table presents that AMNESIAC has poor performance, making it difficult to achieve unlearning in the mixed labels setting, showing high accuracy of $\mathcal{D}_f$ after unlearning. As k increases, it suffers from a significant loss of accuracy on the remaining data $\mathcal{D}_r$. This result again demonstrates that, as mentioned in the paper (Graves, Nagisetty, and Ganesh 2021), AMNESIAC needs fine-tuning with $\mathcal{D}_r$, as a post-processing, to regain performance after the amnesiac unlearning. On the contrary, our approach not only effectively enables the unlearning of $\mathcal{D}_f$, but also maintains the knowledge from $\mathcal{D}_r$ and $\mathcal{D}_f$ when compared to the AMNESIAC unlearning approach. Note that this holds true even as the value of k increases, and all of this is achieved without requiring extra information like model parameter updates.

Results from unlearning age classifiers on UTKFace. For this experiment, we mirror a real-world setting and em-

pirically test unlearning on a classifier that was trained on facial data. Specifically, we first pretrain ResNet-18 on age-group classification on the UTKFace dataset (Zhang, Song, and Qi 2017). UTKFace is a dataset consisted of 20k facial images, each labeled with the age of the subject ranging between 0 and 116. Following previous work Jung et al. (2021), we divide the age range into three (0 to 19, 20 to 40, and more than 40), effectively reducing the task into 3-way classification. After pretraining, we apply instance-wise unlearning to adjust model parameters towards misclassifying k facial images, while aiming to retain its predictive performance on the remaining images.

Table 9 shows the unlearning results. For NEGGRAD, we find that its performance degrades quickly, with its classification performance being lower than that of a random baseline (33% since task is 3-way classification) when unlearning more than 4 instances. On the other hand, both ADV and ADV+IMP consistently outperform the naïve baseline across all k, showing that our regularization techniques are also effective in the facial image domain. The gap between ADV and ADV+IMP is much larger with small k, which demonstrates high potential of weight importance-based regularization for unlearning under a continuous stream of instance-wise data deletion requests.

Details of RAWP

Algorithm 5 presents the pseudocode of RAWP. As an alternative to minimizing cross-entropy loss, we employ adversarial weight perturbation (AWP) (Wu, Xia, and Wang 2020) for our unlearning framework. The main difference between AWP and cross-entropy loss is that AWP does not directly minimize loss through gradient descent, but instead introduces a worst-case weight perturbation with a relative perturbation size constrained by γ. In the manuscript, we reported experimental results using the default value of $\gamma = 0.01$.

Table 7 reports additional experimental results from applying RAWP for unlearning a ResNet-18 pretrained on CIFAR-10 using various γ. We confirm that under various hyperparameterization for γ, RAWP consistently shows performance degradation, especially for large k.

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 139–154.

Aljundi, R.; Rohrbach, M.; and Tuytelaars, T. 2019. Selfless Sequential Learning. In *International Conference on Learning Representations (ICLR)*.

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

Brophy, J.; and Lowd, D. 2021. Machine unlearning for random forests. In *International Conference on Machine Learning*, 1092–1104. PMLR.

Cao, Y.; and Yang, J. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, 463–480. IEEE.

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

Cha, S.; Hsu, H.; Hwang, T.; Calmon, F.; and Moon, T. 2021. CPR: Classifier-Projection Regularization for Continual Learning. In *International Conference on Learning Representations*.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 532–547.

Chundawat, V. S.; Tarun, A. K.; Mandal, M.; and Kankanhalli, M. 2022. Zero-shot machine unlearning. *arXiv preprint arXiv:2201.05629*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Ebrahimi, S.; Meier, F.; Calandra, R.; Darrell, T.; and Rohrbach, M. 2020. Adversarial continual learning. In *European Conference on Computer Vision*, 386–402. Springer.

Eleni Triantafillou, e. a., Fabian Pedregosa. 2023. NeurIPS 2023 Machine Unlearning Challenge.

Ginart, A.; Guan, M.; Valiant, G.; and Zou, J. Y. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32.

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

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Graves, L.; Nagisetty, V.; and Ganesh, V. 2021. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11516–11524.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Heikkilä, M. 2022. What does GPT-3 "know" about me?

Hendrycks, D.; Zhao, K.; Basart, S.; Steinhardt, J.; and Song, D. 2021. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15262–15271.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Iandola, F. N.; Han, S.; Moskewicz, M. W.; Ashraf, K.; Dally, W. J.; and Keutzer, K. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32.

Jung, S.; Ahn, H.; Cha, S.; and Moon, T. 2020. Continual Learning with Node-Importance based Adaptive Group Sparse Regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 3647–3658. Curran Associates, Inc.

Jung, S.; Lee, D.; Park, T.; and Moon, T. 2021. Fair feature distillation for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12115–12124.

Kim, J.; and Woo, S. S. 2022. Efficient Two-Stage Model Retraining for Machine Unlearning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4361–4369.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Kornblith, S.; Norouzi, M.; Lee, H.; and Hinton, G. 2019. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 3519–3529. PMLR.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Kurakin, A.; Goodfellow, I.; Bengio, S.; et al. 2016. Adversarial examples in the physical world.

Lahoti, P.; Beutel, A.; Chen, J.; Lee, K.; Prost, F.; Thain, N.; Wang, X.; and Chi, E. 2020. Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems*, 33: 728–740.

Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; and Graf, H. P. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.

Lison, P.; Pilán, I.; Sánchez, D.; Batet, M.; and Øvrelid, L. 2021. 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)*, 4188–4203.

Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; and Zhang, C. 2017. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, 2736–2744.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mahadevan, A.; and Mathioudakis, M. 2021. Certifiable machine unlearning for linear models. *arXiv preprint arXiv:2106.15093*.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6): 1–35.

Mehta, R.; Pal, S.; Singh, V.; and Ravi, S. N. 2022. Deep Unlearning via Randomized Conditionally Independent Hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10422–10431.

Mirzadeh, S. I.; Chaudhry, A.; Yin, D.; Nguyen, T.; Pascanu, R.; Gorur, D.; and Farajtabar, M. 2022. Architecture matters in continual learning. *arXiv preprint arXiv:2202.00275*.

Molchanov, P.; Mallya, A.; Tyree, S.; Frosio, I.; and Kautz, J. 2019. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11264–11272.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

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

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tarun, A. K.; Chundawat, V. S.; Mandal, M.; and Kankanhalli, M. 2021. Fast yet effective machine unlearning. *arXiv preprint arXiv:2111.08947*.

Toneva, M.; Sordoni, A.; Combes, R. T. d.; Trischler, A.; Bengio, Y.; and Gordon, G. J. 2018. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Villaronga, E. F.; Kieseberg, P.; and Li, T. 2018. Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Law & Security Review*, 34(2): 304–313.

Wen, W.; Wu, C.; Wang, Y.; Chen, Y.; and Li, H. 2016. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29.

Wu, D.; Xia, S.-T.; and Wang, Y. 2020. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33: 2958–2969.

Ye, J.; Fu, Y.; Song, J.; Yang, X.; Liu, S.; Jin, X.; Song, M.; and Wang, X. 2022. Learning with Recoverable Forgetting. *arXiv preprint arXiv:2207.08224*.

Yoon, Y.; Nam, J.; Yun, H.; Kim, D.; and Ok, J. 2022. Few-Shot Unlearning by Model Inversion. *arXiv preprint arXiv:2205.15567*.

Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; and Dwork, C. 2013. Learning fair representations. In *International conference on machine learning*, 325–333. PMLR.

Zhang, Z.; Song, Y.; and Qi, H. 2017. Age Progression/Regression by Conditional Adversarial Autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.