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

Sungmin Cha1*, Sungjun Cho2*, Dasol Hwang2*, Honglak Lee3, Taesup Moon4 and Moontae Lee2, 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 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.

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 is 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 deleting data during training. However, previous work has 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 to uphold privacy rights. This marks a de-
parture from previous approaches that primarily adopted a class-wise unlearning setup towards instance-wise unlearning, 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 requested for deletion. Instead of undoing the previous influence of deleting data, we pursue a stronger goal where all data 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 can be categorized into two approaches: class-wise and instance-wise unlearning. Class-wise unlearning forgets all data points belonging to a certain class (e.g., all images of dogs in CIFAR-10) while retaining performance on the remaining classes (Tarun et al. 2021; Chandawat 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 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 \(D_f\) and the remaining data \(D_r\). For our proposed methods, we require access to no more than the unlearning data \(D_f\), allowing seamless application for practitioners under real-world scenarios. Comparisons with existing unlearning methods are in the supplementary materials.

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; Chen 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).

Method

Preliminaries and Notations

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

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

$$x' = \arg\min_{z: \|z-x\|_p \leq \epsilon} L_{CE}(g_\theta(z), \hat{y}; \theta)$$

where $L_{CE}$ stands for the cross-entropy loss. The $\|z-x\|_p \leq \epsilon$ condition requires that the $L_p$-norm is less than a perturbation budget $\epsilon$. 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 $\Omega$, 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 $L_2$-norm of the outputs:

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

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

---

**Our Proposed Framework**

**Instance-wise unlearning.** Let $\hat{g}_\theta$ 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., $\hat{g}_\theta(x_f) \neq y_f$). (ii) relabeling (or correcting) the predictions of $\mathcal{D}_f$ (i.e., $\hat{g}_\theta(x_f) = y'_f$) where $y'_f \neq y_f$ is chosen individually for each input $x_f$. Let $L_{UL}$ denote the unlearning loss function used to train a classification model. The two goals above can be realized by minimizing the following loss functions:

$$L_{UL}^{MS}(\mathcal{D}_f; \theta) = -L_{CE}(\hat{g}_\theta(x_f), y_f; \theta)$$

$$L_{UL}^{Cor}(\mathcal{D}_f; \theta) = L_{CE}(\hat{g}_\theta(x_f), y'_f; \theta)$$

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: $L_{oracle}(\mathcal{D}_f; \mathcal{D}_r; \theta) = L_{UL}(\mathcal{D}_f; \theta) + L_{CE}(\mathcal{D}_r; \theta)$. 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 $\mathcal{D}_r$:

$$L_{RegUL}(\mathcal{D}_f; \theta) = L_{UL}(\mathcal{D}_f; \theta) + \mathcal{R}(\mathcal{D}_f, g_\theta)$$

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 \( x \) to generate an adversarial example \( x' \) contain class-specific features of the attack target label \( \bar{y} \neq y \).

Based on this finding, we utilize generated adversarial examples as part of regularizer \( R(\cdot) \) to preserve previously learned class-specific knowledge, overcoming forgetting at the representation-level (Algorithm 1 of the supplementary materials). Let \( D_f \) be a set of \( N_f \) images: \( \{(x_f^{(i)}, y_f^{(i)})\}_{i=1}^{N_f} \). Prior to unlearning, we generate adversarial examples \( x_f^{*} \) using the targeted PGD attack with a randomly selected attack target label \( \bar{y} \neq y_f \). We generate \( N_{adv} \) adversarial examples per input \( x_f \), resulting in a set of \( \tilde{N}_f = N_f \times N_{adv} \) examples, denoted as \( \tilde{D}_f = \{(x_f^{(i)}, \bar{y}_f^{(i)})\}_{i=1}^{\tilde{N}_f} \). During unlearning, we add \( \mathcal{L}_{CE}(\tilde{D}_f; \theta) \) with adversarial examples as regularization:

\[
\mathcal{L}_{UL}^{Adv}(D_f; \theta) = \mathcal{L}_{UL}(D_f; \theta) + \mathcal{R}_{Adv}(D_f; g_\theta) \\
= \mathcal{L}_{UL}(D_f; \theta) + \mathcal{L}_{CE}(\tilde{D}_f; \theta) 
\]  

An intuitive illustration of this approach in the representation-level is shown in Figure 1. In essence, the generated adversarial examples in \( D_f \) mimic the remaining dataset \( D_r \), providing information of the pre-trained decision boundary within the representation space. As a result, by adding \( \mathcal{L}_{CE}(\tilde{D}_f; \theta) \) as a regularizer to the unlearning process, the model learns a new decision boundary that minimizes \( \mathcal{L}_{UL} \) (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 \( D_f \) as much as possible, while allowing changes in weights that are considered important for correctly predicting \( D_f \). In other words, we penalize changes of weights that were less important when classifying \( D_f \), thereby preventing weight-level forgetting.

Specifically, we calculate the weight importance with MAS (Aljundi et al. 2018) given \( g_\theta \) and \( D_f \) before unlearning, and normalize the importance measurements \( \Omega^l \) within each \( l \)-th layer to lie within \([0, 1]\). Note that this normalized importance \( \Omega^l \) assigns large values to weights important for classifying \( D_f \). Therefore, we add a regularization that sums all parameter change in the \( l \)-th layer weighted by \( \tilde{\Omega}^l = 1 - \Omega^l \), such that more important weights are updated more. The objective including weight importance with regularization via adversarial examples can be written as:

\[
\mathcal{L}_{UL}^{Adv+Imp}(D_f; \theta) = \mathcal{L}_{UL}^{Adv}(D_f; \theta) + \mathcal{R}_{Imp}(D_f; \theta) \\
= \mathcal{L}_{UL}(D_f; \theta) + \sum_i \tilde{\Omega}_i (\hat{\theta}_i - \tilde{\theta}_i)^2 
\]  

where \( i \) is the index of each weight and \( \hat{\theta} \) 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 \( D_r \), 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 \( D_f \) and negative gradients from \( D_f \) until maximum performance is achieved on \( D_r \) (i.e., \( L_{oracle} \)). In case the accuracy of \( D_f \) does not reach 0 after a certain number of epochs, we report results based on the checkpoint that attained the lowest performance on \( D_f \); NEGGRAD (Golatkar, Achille, and Soatto 2020) fine-tunes the model on \( D_f \) using negative gradients (i.e., \( L_{MS}^{UL} \)) without any regularization; Repeated adversarial weight perturbation (RAPW) is another baseline derived from AWP (Wu, Xia, and Wang 2020) that induces misclassification by adding weight perturbations generated from \( D_f \) to the model parameters. ADV denotes our proposed method using adversarial examples (i.e., \( L_{UL}^{Adv} \)); Lastly, ADV+IMP uses adversarial examples as well as weight importance for regularization (i.e., \( L_{UL}^{Adv+Imp} \)).

**Experimental details.** For each dataset, we randomly pick \( k \in \{16, 64, 128, 256\} \) images from the entire training dataset as unlearning data \( D_f \) and consider the remaining data as \( D_r \). 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 \( D_f \), in case of misclassifying and relabeling, respectively. For generating adversarial examples from \( D_f \), 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 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 1 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 \( D_f \), 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 RAPW in CIFAR-10 when \( k = 256 \)). For
result in a significant loss of accuracy on the remaining data while completely forgetting instances 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.

Table 1: 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 $\mathcal{D}_f$.

Table 2: Instance-wise unlearning results from a ResNet-18 age classifier trained on UTKFace.

In 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 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). Table 2 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 regulariza-
tion 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 3 shows the results of continual unlearning in the model trained with ResNet-50 on CIFAR-100. We observe that NEGRAD 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 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., $\mathcal{L}_{\text{CE}}$). In Table 4, we find that na"{i}ve finetuning attains the best accuracy in both $\mathcal{D}_r$ and $\mathcal{D}_{\text{test}}$ when correcting predictions of a small number of images (e.g. $k = 16$). When correcting larger numbers 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.

### 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.

**A1. Our method shows no pattern in misclassification.** We first check whether the unlearned model classifies all data points 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 NEGRAD, 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

---

**Table 3:** Unlearning instances continually by increments of $k_{CL} = 8$ images per step. Our methods outperform NEGRAD in the continual unlearning scenario as well.

<table>
<thead>
<tr>
<th>$\mathcal{D}_f$ (↓)</th>
<th>BEFORE</th>
<th>NEGGRAD</th>
<th>ADV</th>
<th>ADV+IMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>k = 8</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.56</td>
</tr>
<tr>
<td>k = 16</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.08</td>
</tr>
<tr>
<td>k = 64</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>k = 128</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Table 4:** 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.

<table>
<thead>
<tr>
<th>$\mathcal{D}_f$ (↑)</th>
<th>BEFORE</th>
<th>CORRECT</th>
<th>ADV</th>
<th>ADV+IMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>k = 8</td>
<td>0.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>k = 16</td>
<td>0.0</td>
<td>100.0</td>
<td>95.31</td>
<td>83.44</td>
</tr>
<tr>
<td>k = 32</td>
<td>0.0</td>
<td>100.0</td>
<td>100.0</td>
<td>10.94</td>
</tr>
<tr>
<td>k = 64</td>
<td>0.0</td>
<td>100.0</td>
<td>98.85</td>
<td>9.38</td>
</tr>
<tr>
<td>k = 128</td>
<td>0.0</td>
<td>100.0</td>
<td>98.85</td>
<td>9.38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mathcal{D}_r$ (↑)</th>
<th>BEFORE</th>
<th>CORRECT</th>
<th>ADV</th>
<th>ADV+IMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>k = 8</td>
<td>87.46</td>
<td>87.46</td>
<td>87.46</td>
<td>87.46</td>
</tr>
<tr>
<td>k = 16</td>
<td>84.41</td>
<td>83.29</td>
<td>80.79</td>
<td>77.38</td>
</tr>
<tr>
<td>k = 32</td>
<td>81.75</td>
<td>83.80</td>
<td>83.74</td>
<td>83.44</td>
</tr>
<tr>
<td>k = 64</td>
<td>81.82</td>
<td>83.73</td>
<td>83.53</td>
<td>82.86</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mathcal{D}_{\text{test}}$ (↑)</th>
<th>BEFORE</th>
<th>CORRECT</th>
<th>ADV</th>
<th>ADV+IMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>k = 8</td>
<td>76.15</td>
<td>76.15</td>
<td>76.15</td>
<td>76.15</td>
</tr>
<tr>
<td>k = 16</td>
<td>73.21</td>
<td>72.04</td>
<td>69.91</td>
<td>66.73</td>
</tr>
<tr>
<td>k = 32</td>
<td>70.89</td>
<td>72.58</td>
<td>72.68</td>
<td>72.36</td>
</tr>
<tr>
<td>k = 64</td>
<td>70.98</td>
<td>72.51</td>
<td>72.39</td>
<td>71.68</td>
</tr>
</tbody>
</table>
Figure 2: Confusion matrices from CIFAR-10 showing average pairwise frequencies of pre- (Y-axis) and post-unlearning (X-axis) prediction labels from $D_f$. A blue color indicates higher frequency. Our unlearning framework does not produce any discernible correlation in misclassification.

Figure 3: t-SNE plots of CIFAR-10 datapoints in $D_f$ (triangles) and $D_r$ (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 $D_f$ towards the class boundary to induce misclassification.

Maaten and Hinton (2008) visualizations of final-layer activations from examples in $D_f$ and $D_r$ before and after unlearning. We find that unlearning through only negative gradient significantly distorts the previous decision boundary, leading to poor predictive performance across $D_r$. However, when we incorporate adversarial samples from instances in $D_f$, 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.

**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 $D_f$, 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

This work was done while Sungmin Cha did a research internship at Advanced Machine Learning Lab, LG AI Research.

References


