Stimulating Catastrophic Forgetting in Class-wise Unlearning via UAP

Wenxing Zhou, Xinwen Cheng, Yingwen Wu, Ruikai Yang, and Xiaolin Huang ✉

Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University {apricvivi, xinwencheng, yingwen_wu, ruikai.yang, xiaolinhuang}@sjtu.edu.cn

Abstract. The growing concerns regarding user privacy and data security have brought attention to the task of machine unlearning (MU), which aims to remove the influence of specific data from a well-trained model effectively and efficiently. A naive unlearning method is finetuning the pretrained model to continually learn the remaining data to induce the "catastrophic forgetting" of forgetting data. However, such unlearning often turns out to be inefficient. For effective and efficient unlearning, it is crucial to stimulate catastrophic forgetting, ideally by directly localizing model's knowledge of specific class-wise features associated with the forgetting data. In this paper, we highlight that the targeted universal adversarial perturbation (UAP) implicitly contains class-wise information. In light of this, we propose **Unlearning by UAP ($\mathbf{U^2AP}$)**. By adding the perturbation to clean remaining data during the finetuning process, we shift the model's attention away from the forgetting class directly, stimulating faster and more efficient catastrophic forgetting. Extensive experiments demonstrate that U^2AP enables quicker and more accurate forgetting while maintaining model performance on the remaining data.

Keywords: Machine Unlearning · Machine Learning .

1 Introduction

The success of deep learning is largely driven by the diversity and abundance of training data [42]. However, this success also brings about pressing issues, including data leaks, threats to personal privacy, and misuse of data that may violate regulations [16]. To address these challenges, the General Data Protection Regulation (GDPR) [17] was introduced to ensure "the right to be forgotten" [10]. In response, "machine unlearning" (MU) has emerged as a research area aimed at enabling machine learning models to effectively forget specific data when needed.

In machine unlearning, retraining the model from scratch after data removal remains the golden standard for ensuring complete forgetting [41], yet this approach is computationally intensive. To avoid the need of full retraining, various methods have been proposed. Most of these methods rely on isolating forgetting data influences [14], adjusting network outputs [3], and injecting error messages

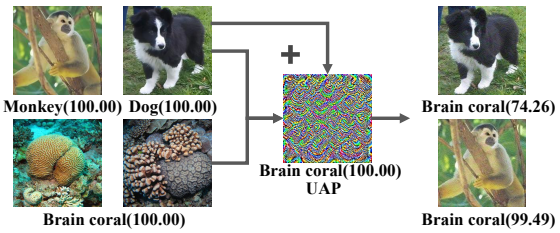

Fig. 1. Explanation of the effect of UAP. The corresponding number after the image class is the predicted probability of being categorized into that class.

to forgetting data to disrupt the network's memory of data [20,9]. However, these methods sometimes compromise both unlearning efficiency and forgetting completeness. Empirical evaluations reveal that such approaches may underperform even compared to naive finetuning. Finetuning trains the pretrained model on the remaining samples, gradually forgetting the unlearned data through catastrophic forgetting [24], providing a milder unlearning process. However, its passive nature results in suboptimal efficiency.

To address this, class-specific features should be involved to actively stimulate catastrophic forgetting. On the one hand, such information should be the key for the well-trained model to recognize the class as being forgotten. On the other hand, it should not be specific to individual images within the same class but rather representative of the entire class. Universal Adversarial Perturbation (UAP) [29], an indistinguishable dark pattern specific to a class, perfectly fits the requirement. When a UAP associated with a specific class is applied to any clean sample, regardless of its original class, the network is deceived into misclassifying the sample as the targeted class. As shown in Fig. 1, when the UAP targeted at "brain coral" is added to clean samples originally classified as "monkey" and "dog", the model's predictions for these samples are fooled to "brain coral" in high probability. These perturbations, when visualized, often contain semantic information of the targeted class. This observation underscores that UAPs encapsulate critical discriminative information that the neural network relies on for decision-making. Consequently, by targeting and eliminating such information during the unlearning process, we can effectively disrupt the model's ability to recognize the forgetting class, thereby stimulating catastrophic forgetting.

Here we propose **Unlearning by UAP ($\mathbf{U^2AP}$)**. By combining the forgetting features with the correct remaining data, our method reduces the information loss associated with relabeling methods and overcomes the inefficiency of simple finetuning. The specific differences between our method and others are illustrated in Fig. 2. When the "brain coral" class is to be forgotten, simple finetuning neglects the forgetting data and gradually forgets "brain coral" by continuing to learn from only the remaining classes. This passive approach, however, proves inefficient, especially when dealing with more complex datasets or networks. To speed up such forgetting, some other prior works generally inject error information into the forgetting data to actively guide the model toward unlearn-

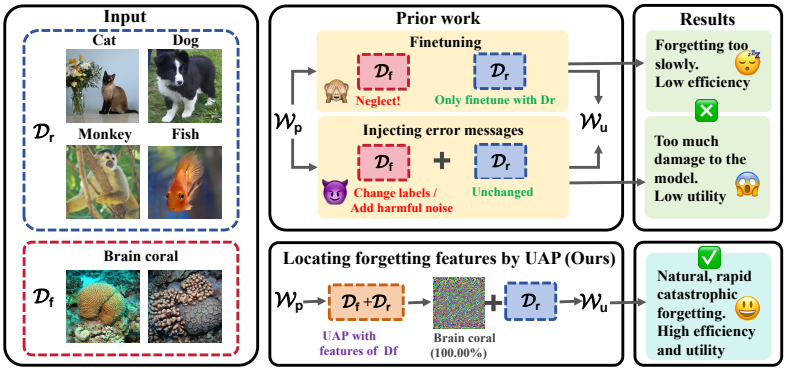

Fig. 2. Comparison of previous unlearning work and ours. $\mathcal{D}_r$ and $\mathcal{D}_f$ denote the remaining and forgetting data, $\mathcal{W}_p$ and $\mathcal{W}_u$ denote the pretrained and unlearned model.

ing specific data. For example, Amnesiac [20] modifies the labels of the forgetting data, and UNSIR [39] adds harmful noise to the forgetting set. However, this kind of error information will also inevitably damage the model's performance in the remaining data, resulting in low utility. Whereas our approach incorporates forgetting features, by applying U^{2}AP, the model's attention is forced to shift from the forgetting class to the remaining data, thus accelerating the erasure of class-specific memories while minimally impacting the remaining set. Like other unlearning methods, the forgetting process affects the remaining data. So we mitigate the potential disruptions to remaining information by finetuning the model on a subset of remaining data after forgetting. This additional step ensures stable performance, enhancing the model's utility. By combining active forgetting with careful post-unlearning refinement, U^{2}AP achieves a balance between efficient unlearning and sustained model performance.

Our contributions can be summarized as follows:

• We explore the implicit representation of the network's memory of specific class-wise features and innovatively extract such information by UAP. By incorporating the UAP into the forgetting process, we localize the memory of the forgetting class, achieving more precise unlearning.

• We propose a novel unlearning method, U^{2}AP. By training targeted UAP and then adding it to the remaining data for finetuning, we stimulate the catastrophic forgetting of the forgetting class, improving the effectiveness and efficiency of class-wise machine unlearning.

• Experimental results across various settings demonstrate that our method accelerates the forgetting process while effectively preserving the model's performance on the remaining data. This improvement is especially prominent when U^{2}AP is applied to larger-scale datasets and more complex models, making it a more efficient and reliable solution for practical applications.

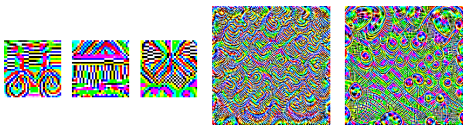

Fig. 3. Results from FG-UAP targeted attacks. The left three shows perturbations on CIFAR-100 using ResNet-18, where each perturbation exhibits visual features specific to the target classes: "bicycle", "house", and "butterfly". Right two displays perturbations on ImageNet-1K using VGG-16, with textures arranged in distinct, patterned forms targeting the classes "brain coral" and "bubble".

2 Related work

2.1 Machine unlearning

Machine unlearning aims to eliminate the influence of specific data on a well-trained model to ensure the legal use of data and the user's privacy [6,13,21]. While retraining is the golden standard, it is resource-intensive, promoting the development of effective and efficient unlearning methods. Current methods mainly achieve unlearning by analyzing data influence through parameter importance selection or stimulating catastrophic forgetting through label modification.

A straightforward approach to analyze and remove data influence is retrieving historical gradients associated with the forgetting data [40]. However, due to the complexity of dynamic training, such retrieval process is inexact and inefficient. Thereby influence function [25] is introduced to approximate data influence for unlearning [21], but is limited to linear models with convex loss functions [2]. More practical influence removal techniques usually rely on parameter importance selection [14], which involves selectively suppressing parameters specifically responsible for the forgetting data.

Other unlearning methods stimulate catastrophic forgetting by finetuning with the modified data. The most basic finetuning trains the pretrained model exclusively on the remaining data, passively relying on the natural process of catastrophic forgetting [24] to gradually erode the model's acquired knowledge of the forgetting data. However, this method is inefficient. To accelerate the forgetting process, some methods inject incorrect information, *e.g.* random-relabeling [20], knowledge-distillation from a useless teacher model [9,27] or assimilating error-maximizing noise [39,8]. However, introducing such incorrect knowledge severely damages model utility, making unlearning inexact, inefficient, and ultimately unfavorable. To achieve more effective and efficient unlearning, it is crucial to correctly locate the knowledge associated with specific class-wise information, thereby stimulating catastrophic forgetting in a targeted and controllable way.

2.2 Universal Adversarial Perturbation

Universal Adversarial Perturbation (UAP) [29] is an adversarial perturbation that can fool a well-trained model to misclassify any sample with the perturba-

tion. In particular, targeted UAP attacks focus on fooling the model's predictions to a specific targeted class, rather than merely causing misclassification [46].

Recently, various UAP methods have been proposed to achieve nearly perfect fooling rates [30,31,47,45]. Interestingly, powerful UAP is revealed to exhibit semantic patterns specific to the targeted class, which are visualized in Fig. 3. These semantic patterns evoke the model's response and successfully fool its predictions on perturbed data. This demonstrates that it is possible to extract the specific information related to a targeted class from a pretrained model through UAP [32,35,7]. Such insights provide a method for locating the class-wise knowledge embedded in the model, which can then be leveraged to facilitate effective unlearning.

3 Method

3.1 Preliminaries

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be a training set consisting of samples $x_i \in \mathbb{R}^d$ and corresponding class labels $y_i \in \{1, ..., K\}$. Machine unlearning aims to eliminate the influence of specific samples that need to be forgotten from a pretrained model. These samples are referred to as the *forgetting data* and denoted as $\mathcal{D}_f = \{(x_i, y_i) \in \mathcal{D}\}_{i=1}^{N_f}$. The remaining samples in the dataset form the *remaining set*, denoted as $\mathcal{D}_r = \mathcal{D} \backslash \mathcal{D}_f$. The unlearned model should be as close as possible to the retrained model, which is trained from scratch with $\mathcal{D}_r$.

Nowadays, machine unlearning includes three primary tasks: (1) Full-class unlearning: where the $\mathcal{D}_f$ is made of all samples from an entire class $k \in \{1, ..., K\}$, (2) Sub-class unlearning: where the $\mathcal{D}_f$ is made of samples from a specific sub-class under a broader super-class $k \in \{1, ..., K\}$, (3) Random sample unlearning: where the $\mathcal{D}_f$ is made of samples randomly chosen from the entire dataset $\mathcal{D}$. Also, we consider multi-class unlearning: where the $\mathcal{D}_f$ is made of all samples from several entire classes $k_1, k_2, ... \in \{1, ..., K\}$. Here we mainly focus on full-class unlearning, sub-class unlearning, and multi-class unlearning.

3.2 Proposed method

In deep learning, catastrophic forgetting refers to the phenomenon where neural networks progressively lose previously acquired knowledge when sequentially trained on new tasks. Recent studies have leveraged this property through finetuning-based unlearning methods to eliminate information pertaining to specific forgetting data. However, continual learning on the remaining data may fail to achieve effective forgetting due to the underlying resemblance between the remaining and the forgetting data. To address this challenge, we propose that effective machine unlearning should actively stimulate controllable catastrophic forgetting through targeted interference with class-specific features. The most straightforward approach is to extract the pretrained model's knowledge of the forgetting data and remove it in a targeted way. Our key insight stems from the

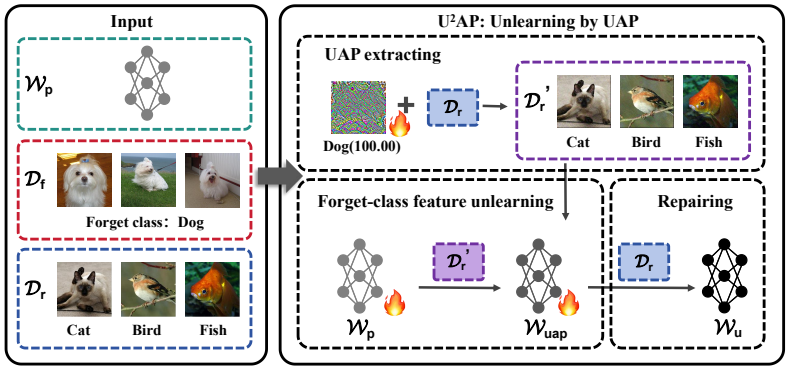

Fig. 4. The specific process of the three steps of U^{2}AP, taking the example of forgetting "dog". (1) UAP extracting: using FG-UAP to targeted attack pretrained model in remaining data to get targeted UAP of "dog", (2) forget-class feature unlearning: adding the UAP to remaining data and finetuning the pretrained model with correct remaining data labels to stimulate catastrophic forgetting, (3) repairing: finetuning model with subset of clean remaining data to repair model utility to get final unlearned model.

observation that UAPs generated by pretrained models inherently encapsulate discriminative semantic patterns of targeted classes, as evidenced by their class-specific visual manifestations in Fig. 3, which indicates the pretrained model's memorization of the classes. This suggests that UAPs can serve as effective proxies for extracting class-wise information stored in model parameters. In light of this, we propose a method that first uses UAP to extract class-wise information and then promotes the shift of the model's attention by training on the UAP-perturbed remaining data. The labels of these perturbed samples still align with those of their clean counterparts, compelling the model to actively ignore the acquired knowledge of the targeted forgetting class. To mitigate any potential impact of the adversarial noise on model utility, we finetune the model with the clean remaining data for a few steps following previous methods. Overall, U^{2}AP has three steps: UAP extracting, forget-class feature unlearning, and model performance repairing. The specific framework of our method is illustrated in Fig. 4.

UAP extracting. Targeted UAP is an input-agnostic noise that induces misclassification of all perturbed samples toward a predefined target class. Interestingly, targeted UAP is revealed to contain semantic patterns specific to the targeted class. This behavior demonstrates that UAPs inherently encode a model's memorization of class-specific information. Therefore, it can be an effective tool for removing the influence of forgetting data from pretrained model. In this paper, we use FG-UAP [45] for its obvious class-wise patterns. As revealed by Papyan *et al.* [34], for samples belonging to the same class, their penultimate features in a well-trained model exhibit totally identical. FG-UAP trains the UAP to destroy such within-class feature collapse of perturbed data. By enforcing the classification of perturbed data to the targeted class with a cross-entropy loss, FG-UAP achieves a remarkable fooling rate. Mathematically, the FG loss

is expressed as:

$$\mathcal{L}_{\mathrm{FG}}(\mathcal{D}_r, \delta, y_f) = \sum_{x_r \in \mathcal{D}_r} [\mathcal{L}_{\mathrm{attack}}(x_r, \delta) + \mathcal{L}_{\mathrm{CE}}(x_r + \delta, y_f)] \tag{1}$$

where

$$\mathcal{L}_{\mathrm{attack}}(x_r, \delta) = \frac{h(x_r) \cdot h(x_r + \delta)}{\|h(x_r)\| \|h(x_r + \delta)\|} \tag{2}$$

and $h(x)$ represents the feature of sample x, δ is the UAP needs to be optimized, y_f denotes the forgetting class. We use the $\mathcal{L}_{\mathrm{attack}}$ to optimize the UAP and use the $\mathcal{L}_{\mathrm{CE}}$ to targetedly drive the predictions of the perturbed samples to the forgetting class.

Forget-class feature unlearning. The targeted UAP inherently encodes the class-wise information that establishes strong parametric interactions with the pretrained model's weight space. This causes the pretrained model to focus heavily on this class-wise information once it is introduced to clean samples. Therefore, in the context of unlearning, the goal is to reduce the model's attention to such features, *i.e.*, to make the model ignore and diminish its responsiveness to them. To achieve this, we finetune the pretrained model to correctly classify the perturbed remaining data to its original labels with the loss function in Equation (3). The inclusion of these perturbed samples causes the model's attention to shift rapidly away from the targeted UAP, thereby actively stimulating the process of "catastrophic forgetting" by overriding the explicit semantic information embedded in the UAP.

$$\mathcal{L}_{\mathrm{forget}}(\mathcal{D}_r, \delta) = \sum_{x_r, y_r \in \mathcal{D}_r} \mathcal{L}_{\mathrm{CE}}(x_r + \delta, y_r) \tag{3}$$

Repairing. Since the UAP is essentially an adversarial noise, learning the perturbed data may potentially impact performance on remaining data. Fortunately, this can be effectively mitigated by finetuning the model for a few steps on clean samples, as other unlearning methods also explicitly or implicitly leverage remaining data for repairing [39,9,20,14]. For full-class unlearning, we repair by finetuning on a subset of remaining data. For sub-class unlearning, detailed sub-class labels within broader super-classes can help target the UAP specifically for certain classes. However, when sub-class labels are inaccessible, as in the MU setting, we generate the UAP by targeted attacking the entire super-class and then use a subset of remaining data for repairing.

The complete unlearning algorithm, which consists of the three critical steps outlined above: (1) UAP extracting, (2) forget-class feature unlearning, (3) repairing, is presented in Algorithm 1.

4 Experiments

4.1 Setups

Datasets, models and MU tasks. We evaluate our U^{2}AP in supervised image classification tasks. Since we utilize the UAP to extract class-wise information,

Algorithm 1 Unlearning by UAP (U^2AP)

Input: Pretrained model $\mathcal{W}_p$, Remaining set $\mathcal{D}_r$, Forgetting set $\mathcal{D}_f$, Forgetting class y_f
Output: Unlearned model $\mathcal{W}_u$
Step 1: UAP Extracting
Sample subset $\mathcal{D}_{r1}$ from $\mathcal{D}_r$
$\delta \leftarrow \text{FG\_UAP}(\mathcal{W}_p, \mathcal{D}_{r1}, y_f)$
Step 2: Forget-Class Feature Unlearning
Sample subset $\mathcal{D}_{r2}$ from $\mathcal{D}_r$ based on UAP
Initialize synthetic dataset $\mathcal{D}_s = \{\}$
for $x_i \in \mathcal{D}_{r2}$ **do**
    Add UAP to sample: $x_i' = x_i + \delta$
    Add (x_i', y_i) to $\mathcal{D}_s$
    $\mathcal{W}_{uap} \leftarrow$ Finetune ($\mathcal{W}_p$, $\mathcal{D}_s$)
end for
Step 3: Repairing
Sample subset $\mathcal{D}_{r3}$ from $\mathcal{D}_r$
for x_i in $\mathcal{D}_{r3}$ **do**
    $\mathcal{W}_u \leftarrow$ Finetune ($\mathcal{W}_{uap}$, $\mathcal{D}_{r3}$)
end for
Return: Unlearned model $\mathcal{W}_u$

we focus on class-wise unlearning tasks, including full-class unlearning, sub-class unlearning, and multi-class unlearning. The full-class unlearning is evaluated on ImageNet-1K [11]. The sub-class unlearning is evaluated on sub-classes of CIFAR-20 [26]. The multi-class unlearning is evaluated on several classes of CIFAR-100 [26]. For smaller-scale datasets like CIFAR-100 and CIFAR-20, we use ResNet-18 [22] as the training model, while for the larger-scale dataset, ImageNet-1K, we use Deit-B [43]. In our experiments, the forgotten classes were selected randomly.

Evaluation metrics. Machine unlearning algorithms should be evaluated based on their ability to effectively remove information while maintaining model performance and ensuring efficiency. Our evaluation metrics focus on four critical aspects: the effectiveness of forgetting, the utility of model, the protection of privacy, and the efficiency of the method. For the effectiveness of forgetting, we use the forgetting accuracy (**FA**) to measure how well the model has removed the influence of forgetting data. For the utility of model, we use the remaining accuracy (**RA**) and validation accuracy (**VA**) to evaluate how well the model retains performance on the remaining data. To provide a more comprehensive evaluation, we compute the average gap (**Avg. Gap**) between the unlearned model and the retrained model across **FA**, **RA** and **VA**. A smaller average gap indicates better unlearning performance. Additionally, for the protection of privacy, we employ the membership inference attacks (**MIA**) [37,38] to further evaluate the privacy guarantees after unlearning. The success rate of MIA indicates how many samples in $\mathcal{D}_f$ are classified as member samples of the unlearned model. The lower MIA represents less information about the forgetting set left in the

unlearned model, indicating more effective forgetting. Regarding the efficiency of the unlearning algorithm, we report the execution time (**Time**) of each method.

Baselines. We compare our method against eight baselines, including (1) Finetune: training the pretrained model with remaining data to gradually unlearn through catastrophic forgetting [44], (2) Gradient Ascent: unlearning by making a gradient ascent on the forgetting data [40], (3) EU-k and (4) CF-k: fixing some parameters and only finetuning the last k layers [18], (5) Unlearning-by-Selective-Impair-and-Repair (UNSIR): generating noise to impair the forgetting data and then finetining with the subset of remaining data to repair the model [39], (6) Bad Teacher: transferring knowledge from the useful (the pretrained model) and useless (the randomly initialized model) teachers for the remaining data and the forgetting data [9], (7) Amnesiac: relabeling the forgetting data and meanwhile minimizing the cross-entropy loss of the remaining data [20], (8) Saliency-based Unlearning (SalUn): computing a weight saliency map to identify the weights most relevant to forgetting data by gradient norm [14].

Table 1. Results of full-class unlearning methods on ImageNet using Deit-B, evaluated with two classes "109" an "971". Each value represents the $\text{mean}_{\pm\text{variance}}$ in percent(%). (Avg. Gap with retraining is not computed here because retraining is too time-consuming.) We bold the best-performing values.

Class	Method	**RA↑**	**FA↓**	**VA↑**	**MIA↓**	**Time (s)↓**
109	Pretrain	$81.89_{\pm 0.00}$	$88_{\pm 0.00}$	$80.10_{\pm 0.00}$	$63.50_{\pm 0.00}$	-
	Finetune	$72.95_{\pm 0.45}$	$4.82_{\pm 3.03}$	$71.94_{\pm 0.44}$	$23.50_{\pm 6.38}$	9171
	GradientAscent	$75.76_{\pm 0.21}$	$15.67_{\pm 0.34}$	$73.44_{\pm 0.33}$	$24.56_{\pm 0.76}$	5130
	EU-k	$73.56_{\pm 0.33}$	$5.68_{\pm 0.23}$	$71.31_{\pm 0.26}$	$10.51_{\pm 1.75}$	8451
	CF-k	$75.66_{\pm 0.61}$	$6.52_{\pm 1.09}$	$72.89_{\pm 0.38}$	$7.66_{\pm 0.10}$	7989
	UNSIR	$68.17_{\pm 1.32}$	$49.03_{\pm 1.00}$	$65.14_{\pm 1.77}$	$56.12_{\pm 0.88}$	6513
	BadTeacher	$70.04_{\pm 0.50}$	$8.33_{\pm 9.83}$	$69.95_{\pm 0.45}$	$2.33_{\pm 1.52}$	4934
	Amnesiac	$78.57_{\pm 0.21}$	$9.32_{\pm 3.52}$	$78.54_{\pm 0.21}$	$\mathbf{1.33_{\pm 0.58}}$	5225
	SalUn	$79.58_{\pm 0.05}$	$1.04_{\pm 1.00}$	$79.53_{\pm 0.05}$	$2.33_{\pm 1.04}$	4840
	U^{2}AP	$\mathbf{80.25_{\pm 0.13}}$	$\mathbf{0.00_{\pm 0.00}}$	$\mathbf{80.20_{\pm 0.12}}$	$4.00_{\pm 0.50}$	**748**
971	Pretrain	$81.73_{\pm 0.00}$	$85.42_{\pm 0.00}$	$81.72_{\pm 0.00}$	$79.50_{\pm 0.00}$	-
	Finetune	$72.12_{\pm 0.37}$	$5.20_{\pm 3.89}$	$71.10_{\pm 0.37}$	$11.83_{\pm 3.05}$	9197
	GradientAscent	$76.21_{\pm 0.98}$	$8.73_{\pm 0.52}$	$74.08_{\pm 1.55}$	$9.76_{\pm 0.20}$	5326
	EU-k	$77.98_{\pm 1.62}$	$7.36_{\pm 0.54}$	$75.10_{\pm 0.91}$	$6.78_{\pm 2.81}$	8030
	CF-k	$76.65_{\pm 1.00}$	$8.52_{\pm 0.83}$	$73.87_{\pm 0.20}$	$8.98_{\pm 0.55}$	8006
	UNSIR	$72.98_{\pm 0.97}$	$50.31_{\pm 0.84}$	$70.45_{\pm 1.02}$	$47.89_{\pm 0.76}$	6002
	BadTeacher	$70.27_{\pm 0.41}$	$13.33_{\pm 5.16}$	$70.21_{\pm 0.43}$	$1.83_{\pm 0.29}$	4929
	Amnesiac	$78.70_{\pm 0.12}$	$16.28_{\pm 5.66}$	$78.66_{\pm 0.12}$	$\mathbf{1.33_{\pm 0.29}}$	5251
	SalUn	$79.59_{\pm 0.08}$	$7.00_{\pm 2.76}$	$79.55_{\pm 0.08}$	$2.17_{\pm 0.58}$	4774
	U^{2}AP	$\mathbf{80.11_{\pm 0.04}}$	$\mathbf{0.00_{\pm 0.00}}$	$\mathbf{80.07_{\pm 0.04}}$	$3.17_{\pm 0.76}$	**525**

4.2 Results and comparison

Performace on full-class unlearning tasks. The advantages of our approach become noticeable when the models and datasets are more complex and large-scaled. For the ImageNet-1K dataset shown in Table 1, the non-zero values of FA of other baselines indicate that they all fail to achieve complete forgetting without unduly impairing model performance. Although the FAs may drop to 0 as the training duration increases, it will inevitably cause lower RA. In contrast, U^{2}AP achieves a zero-value FA as well as high RA and VA. Furthermore, our method achieves low MIA scores, which reflects its enhanced privacy protection capabilities after unlearning. In terms of forgetting efficiency, the time required by our method is significantly lower (faster $\times 10$) than that of other baselines. Especially, compared to basic finetuning where catastrophic forgetting is inefficient in this task, our method significantly accelerates such process. Other baseline approaches not only fail to achieve complete forgetting knowledge elimination but also induce severe performance degradation in model utility. Unlike other methods that suffer from excessively long execution times and low efficiency, our method proves more practical for real-world applications.

Table 2. The time(s) consumption of each step in the U^{2}AP framework: (1) UAP extracting; (2) Forget-class feature unlearning; (3) Repairing. And the accuracy on different datasets after forgetting step. The results are evaluated on "109" and "971" classes of ImageNet-1K using Deit-B.

Class	$\mathbf{RA}_u \uparrow$	$\mathbf{FA}_u \downarrow$	$\mathbf{VA}_u \uparrow$	**UAP Time**	**Unlearn Time**	**Repair Time**
109	$78.21_{\pm 0.63}$	$0.00_{\pm 0.00}$	$76.87_{\pm 0.43}$	102	160	402
971	$77.89_{\pm 0.25}$	$0.00_{\pm 0.00}$	$76.35_{\pm 0.57}$	101	161	250

To comprehensively evaluate the efficiency of our proposed method under complex configurations of datasets and models, we conducted detailed experiments to measure the time consumption of each step in the U^{2}AP framework on ImageNet-1K. Furthermore, to demonstrate that UAP specifically stimulates catastrophic forgetting for the targeted class while minimally affecting other classes, we systematically measured model accuracy across all datasets immediately after the forgetting step (prior to the repairing step), as presented in Table 2. Our results show that the computation time required for UAP remains acceptable given the complexity of both models and datasets, with the forgetting step itself requiring negligible time. Remarkably, this efficient forgetting procedure sufficiently reduces the FA value to 0% while maintaining high accuracy on remaining data (exhibiting less than 4% degradation compared to pre-forgetting performance). These results confirm that UAP achieves efficient elimination of target class influence while preserving model utility for other classes, thereby validating the operational efficiency and precision of our approach.

Table 3. Results of multi-class unlearning methods on CIFAR-100 using ResNet-18, evaluated with forgetting 2, 4, 8 classes. Since the purpose of forgetting the whole class is high RA,VA as well as low FA, we omit the comparison with retrain here. The tables are laid out in the same format as Table 1.

Classes number	Method	**RA↑**	**FA↓**	**VA↑**	**MIA↓**
2	Pretrain	$76.28_{\pm 0.00}$	$83.33_{\pm 0.00}$	$76.50_{\pm 0.00}$	$95.10_{\pm 0.00}$
	Finetune	$76.43_{\pm 0.22}$	$0.02_{\pm 0.05}$	$70.11_{\pm 0.18}$	$1.58_{\pm 0.25}$
	GradientAscent	$73.01_{\pm 0.64}$	$1.20_{\pm 1.30}$	$71.54_{\pm 0.40}$	$7.23_{\pm 0.34}$
	EU-k	$68.53_{\pm 0.60}$	$0.00_{\pm 0.00}$	$66.96_{\pm 0.56}$	$3.00_{\pm 0.03}$
	CF-k	$72.76_{\pm 0.22}$	$0.00_{\pm 0.00}$	$71.57_{\pm 0.20}$	$2.13_{\pm 0.01}$
	BadTeacher	$66.86_{\pm 0.46}$	$0.00_{\pm 0.00}$	$65.47_{\pm 0.44}$	$\mathbf{0.00_{\pm 0.00}}$
	Amnesiac	$73.39_{\pm 0.60}$	$0.00_{\pm 0.00}$	$71.93_{\pm 0.60}$	$46.62_{\pm 1.04}$
	SalUn	$76.50_{\pm 0.14}$	$2.68_{\pm 0.97}$	$\mathbf{74.99_{\pm 0.13}}$	$\mathbf{0.00_{\pm 0.00}}$
	U^{2}AP	$\mathbf{76.62_{\pm 0.06}}$	$\mathbf{0.00_{\pm 0.00}}$	$74.95_{\pm 0.23}$	$1.11_{\pm 0.39}$
4	Pretrain	$76.21_{\pm 0.00}$	$80.66_{\pm 0.00}$	$76.50_{\pm 0.00}$	$95.25_{\pm 0.00}$
	Finetune	$75.91_{\pm 0.14}$	$0.00_{\pm 0.00}$	$72.93_{\pm 0.54}$	$3.83_{\pm 6.89}$
	GradientAscent	$73.54_{\pm 0.20}$	$3.51_{\pm 0.34}$	$71.92_{\pm 0.33}$	$2.81_{\pm 0.56}$
	EU-k	$68.32_{\pm 0.57}$	$0.06_{\pm 0.11}$	$67.13_{\pm 0.35}$	$4.13_{\pm 0.02}$
	CF-k	$72.98_{\pm 0.31}$	$0.00_{\pm 0.00}$	$71.62_{\pm 0.38}$	$1.66_{\pm 0.20}$
	BadTeacher	$67.09_{\pm 0.52}$	$0.04_{\pm 0.09}$	$64.35_{\pm 0.50}$	$\mathbf{0.00_{\pm 0.00}}$
	Amnesiac	$72.81_{\pm 0.12}$	$0.00_{\pm 0.00}$	$69.83_{\pm 0.14}$	$44.33_{\pm 0.60}$
	SalUn	$76.14_{\pm 0.19}$	$1.76_{\pm 0.69}$	$73.11_{\pm 0.17}$	$0.01_{\pm 0.02}$
	U^{2}AP	$\mathbf{76.50_{\pm 0.33}}$	$\mathbf{0.00_{\pm 0.00}}$	$\mathbf{73.36_{\pm 0.34}}$	$1.02_{\pm 0.35}$
8	Pretrain	$76.13_{\pm 0.00}$	$80.80_{\pm 0.00}$	$76.50_{\pm 0.00}$	$95.82_{\pm 0.00}$
	Finetune	$76.43_{\pm 0.22}$	$0.02_{\pm 0.05}$	$70.11_{\pm 0.18}$	$1.58_{\pm 0.25}$
	GradientAscent	$74.06_{\pm 0.72}$	$3.61_{\pm 0.60}$	$72.36_{\pm 0.33}$	$9.22_{\pm 1.65}$
	EU-k	$69.06_{\pm 0.67}$	$0.00_{\pm 0.00}$	$66.34_{\pm 0.66}$	$2.02_{\pm 0.01}$
	CF-k	$73.21_{\pm 0.89}$	$1.05_{\pm 0.09}$	$\mathbf{71.00_{\pm 0.66}}$	$0.56_{\pm 0.93}$
	BadTeacher	$67.35_{\pm 0.48}$	$0.00_{\pm 0.00}$	$61.84_{\pm 0.47}$	$\mathbf{0.00_{\pm 0.00}}$
	Amnesiac	$72.67_{\pm 0.25}$	$0.00_{\pm 0.00}$	$66.74_{\pm 0.24}$	$43.71_{\pm 0.94}$
	SalUn	$75.65_{\pm 0.20}$	$3.17_{\pm 0.41}$	$69.75_{\pm 0.22}$	$0.03_{\pm 0.03}$
	U^{2}AP	$\mathbf{76.54_{\pm 0.13}}$	$\mathbf{0.00_{\pm 0.00}}$	$70.87_{\pm 0.22}$	$2.14_{\pm 0.32}$

Performace on multi-class unlearning tasks. To further analyze the applicability and stability of our method across different unlearning scenarios, we conducted multi-class unlearning experiments with varying numbers of forgetting classes (2, 4, and 8) on CIFAR-100 using ResNet-18. We train the corresponding UAPs for the classes that need to be forgotten and add them randomly to the remaining data (only one class of UAPs is added to each remaining sample). As shown in Table 3, our method consistently achieves optimal performance across all evaluation metrics, demonstrating superior RA and VA while maintaining the lowest FA. Comparative analysis reveals that while Finetune and SalUn preserve model utility on remaining classes, they exhibit incomplete forgetting of target classes. Conversely, EU-k, CF-k, and Amnesiac achieve effective forgetting but significantly degrade performance in the remaining classes. Our approach effectively balances these objectives, delivering both precise forgetting and high utility preservation without compromising either aspect.

Performace on sub-class unlearning tasks. We do experiments of sub-class "rocket" on CIFAR-20. As shown in Table 4, data-modification based methods like Amnesiac and UNSIR both have relatively RA, indicating poor forgetting effectiveness. Some other finetuning-based methods *e.g.* Gradient Ascent, EU-k, and Cf-k demonstrate performance substantially deviating from retraining. In contrast, in terms of Avg. Gap, our method produces results similar to retraining, with comparatively lower execution time and stable performance.

Table 4. Results of sub-class unlearning methods on CIFAR-20 using ResNet-18, evaluated with sub-class "rocket". The tables are laid out in the same format as Table 1.

Method	**RA↑**	**FA↓**	**VA↑**	**Avg. Gap↓**	**MIA↓**	**Time (s)↓**
Pretrain	$85.26_{\pm 0.00}$	$80.73_{\pm 0.00}$	$85.21_{\pm 0.00}$	-	$92.80_{\pm 0.00}$	-
Retrain	$84.85_{\pm 0.05}$	$2.69_{\pm 0.21}$	$84.07_{\pm 0.09}$	-	$12.40_{\pm 0.86}$	-
Finetune	$83.23_{\pm 0.20}$	$4.46_{\pm 1.44}$	$82.49_{\pm 0.20}$	1.66	$4.36_{\pm 0.92}$	165
GradientAscent	$80.77_{\pm 4.04}$	$1.32_{\pm 1.54}$	$79.53_{\pm 1.28}$	3.33	$14.76_{\pm 5.39}$	130
EU-k	$78.56_{\pm 1.25}$	$1.89_{\pm 0.36}$	$77.97_{\pm 0.26}$	4.40	$16.98_{\pm 2.76}$	230
CF-k	$80.13_{\pm 0.78}$	$3.96_{\pm 1.22}$	$79.04_{\pm 0.88}$	3.67	$9.67_{\pm 0.77}$	254
UNSIR	$78.39_{\pm 0.66}$	$4.69_{\pm 3.23}$	$77.65_{\pm 0.63}$	4.96	$12.08_{\pm 4.85}$	88
BadTeacher	$84.15_{\pm 0.33}$	$3.01_{\pm 1.62}$	$83.34_{\pm 0.32}$	0.58	$\mathbf{0.00_{\pm 0.00}}$	33
Amnesiac	$82.92_{\pm 0.09}$	$\mathbf{2.71_{\pm 1.09}}$	$82.15_{\pm 0.10}$	1.29	$1.10_{\pm 0.48}$	28
SalUn	$\mathbf{84.71_{\pm 0.09}}$	$3.52_{\pm 0.57}$	$\mathbf{83.89_{\pm 0.08}}$	0.38	$0.30_{\pm 0.12}$	117
U^{2}AP	$84.38_{\pm 0.04}$	$2.56_{\pm 0.55}$	$83.56_{\pm 0.04}$	**0.37**	$1.31_{\pm 1.56}$	102

Summary. These results indicate that our method demonstrates consistent forgetting performance across datasets, models of various sizes, and different unlearning tasks. It enables the unlearning process to stimulate the catastrophic forgetting of data while preserving the integrity on remaining data. This makes our approach more effective and efficient than existing unlearning methods.

4.3 Analysis

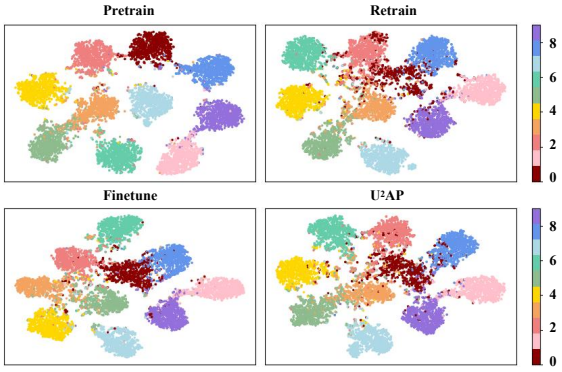

Fig. 5. Visualization of feature dimension reduction for different models on the CIFAR-10. Compared to finetuning, our method has more similar results to retraining.

Feature representation visualization. We use T-SNE [28] dimensionality reduction to visualize the output features of CIFAR-10 [26] across models. In this case, class "0" is the forgetting class, indicated in dark red color. As can be seen in Fig. 5, although both Finetune and our method can distinguish the forgetting data from the remaining data, our approach demonstrates superior feature distribution alignment with the retrained model.

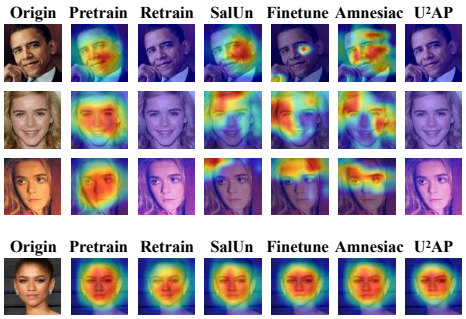

Fig. 6. The visualizations of attention heatmaps for some unlearned models on the forgetting data (top three lines) and the remaining data (bottom line) .

Attention maps visualization. We examine the model's attention on the forgetting class before and after unlearning using attention heatmaps [36]. We compare U^2AP with several other unlearning methods that exhibit minimal Avg. Gap to the retrained model in previous evaluations, including SalUn, Finetune,

and Amnesiac on PinsFaceRecognition [5] using ResNet-18. As shown in Fig. 6, after applying U^{2}AP for class forgetting, the model's attention noticeably shifts away from facial regions, closely aligning with the visualization results of the retrained model. In contrast, the attention maps of other baseline methods still focus more or less on the facial areas. This implies that U^{2}AP is superior to other methods in precisely locating and erasing class-wise information, leading to a more effective unlearning process. In addition, Fig. 6 shows the heatmap of the remaining data before and after forgetting, and it can be seen that the attention of the model is still gathered on the face of the remaining data after forgetting using our method, showing the high utility of our method.

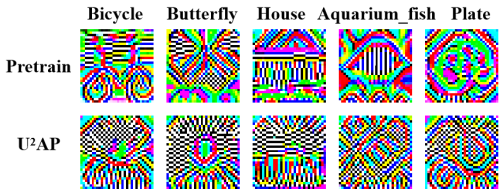

Fig. 7. The UAPs obtained from targeted attacking the model before and after unlearning by U^{2}AP.

Post-unlearned UAP test. We investigate the leftover semantic information of the forgetting class after unlearning by visualizing the targeted UAPs of the pretrained and unlearned models. As shown in Fig. 7, we analyze this for five different forgetting classes in CIFAR-100. The first row shows the UAPs trained from the pretrained model, where each UAP exhibits obvious semantic class-wise patterns, such as a "bicycle" or a "butterfly". After applying U^{2}AP, the semantic patterns disappear from generated UAPs. This demonstrates that our proposed method, U^{2}AP, successfully eliminates the class-wise information from the pretrained model.

5 Conclusion

In this paper, we point out that effective and efficient unlearning requires explicitly extracting class-wise information to stimulate catastrophic forgetting. We emphasize that targeted universal adversarial perturbations implicitly extract class-specific information, as demonstrated by the visible semantic patterns aligned with the targeted class. Hence, the targeted UAP generated from the pretrained model can be readily leveraged to facilitate the unlearning of specific data. In light of this, we propose U^{2}AP, which unlearns by finetuning the model with perturbed remaining data to stimulate catastrophic forgetting. Extensive experiments demonstrate the effectiveness and efficiency of our proposed method. Taking advantage of the universal adversarial perturbation opens up a new perspective for identifying model's knowledge of the forgetting data.

Limitations. The critical aspect of our U^{2}AP lies in extracting implicit features related to the forgetting data, which makes our method dependent on the difficulty and accuracy of feature localization. When the features of the forgetting data are dispersed and difficult to extract, our method may fail. Additionally, we only focus on image tasks, and our approach remains to be explored for tasks such as text-based ones.

Acknowledgments. The research leading to these results has received funding from National Natural Science Foundation of China (62376155) and the Interdisciplinary Program of Shanghai Jiao Tong University (No. YG2022ZD031).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., Lacoste-Julien, S.: A closer look at memorization in deep networks. In: International conference on machine learning. pp. 233–242. PMLR (2017)
2. Bae, J., Ng, N., Lo, A., Ghassemi, M., Grosse, R.B.: If influence functions are the answer, then what is the question? Advances in Neural Information Processing Systems **35**, 17953–17967 (2022)
3. Baumhauer, T., Schöttle, P., Zeppelzauer, M.: Machine unlearning: Linear filtration for logit-based classifiers. Machine Learning **111**(9), 3203–3226 (2022)
4. Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C.A., Jia, H., Travers, A., Zhang, B., Lie, D., Papernot, N.: Machine unlearning. In: 2021 IEEE Symposium on Security and Privacy (SP). pp. 141–159. IEEE (2021)
5. Burak: Pinterest face recognition dataset. www.kaggle.com/datasets/hereisburak/pins-facerecognition (2020)
6. Cao, Y., Yang, J.: Towards making systems forget with machine unlearning. In: 2015 IEEE symposium on security and privacy. pp. 463–480. IEEE (2015)
7. Chang, Y., Ren, Z., Nguyen, T.T., Nejdl, W., Schuller, B.W.: Example-based explanations with adversarial attacks for respiratory sound analysis. arXiv preprint arXiv:2203.16141 (2022)
8. Choi, D., Choi, S., Lee, E., Seo, J., Na, D.: Towards efficient machine unlearning with data augmentation: Guided loss-increasing (gli) to prevent the catastrophic model utility drop. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 93–102 (2024)
9. Chundawat, V.S., Tarun, A.K., Mandal, M., Kankanhalli, M.: Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 7210–7217 (2023)
10. Dang, Q.V.: Right to be forgotten in the age of machine learning. In: Advances in Digital Science: ICADS 2021. pp. 403–411. Springer (2021)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

12. Dukler, Y., Bowman, B., Achille, A., Golatkar, A., Swaminathan, A., Soatto, S.: Safe: Machine unlearning with shard graphs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17108–17118 (2023)
13. Dwork, C.: Differential privacy. In: International colloquium on automata, languages, and programming. pp. 1–12. Springer (2006)
14. Fan, C., Liu, J., Zhang, Y., Wei, D., Wong, E., Liu, S.: Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. arXiv preprint arXiv:2310.12508 (2023)
15. Foster, J., Schoepf, S., Brintrup, A.: Fast machine unlearning without retraining through selective synaptic dampening. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 12043–12051 (2024)
16. Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., Ristenpart, T.: Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing. In: 23rd USENIX security symposium (USENIX Security 14). pp. 17–32 (2014)
17. GDPR, G.D.P.R.: General data protection regulation. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (2016)
18. Goel, S., Prabhu, A., Sanyal, A., Lim, S.N., Torr, P., Kumaraguru, P.: Towards adversarial evaluations for inexact machine unlearning. arXiv preprint arXiv:2201.06640 (2022)
19. Goldman, E.: An introduction to the california consumer privacy act (ccpa). Santa Clara Univ. Legal Studies Research Paper (2020)
20. Graves, L., Nagisetty, V., Ganesh, V.: Amnesiac machine learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11516–11524 (2021)
21. Guo, C., Goldstein, T., Hannun, A., Van Der Maaten, L.: Certified data removal from machine learning models. arXiv preprint arXiv:1911.03030 (2019)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
23. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. Advances in neural information processing systems **32** (2019)
24. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
25. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: International conference on machine learning. pp. 1885–1894. PMLR (2017)
26. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
27. Kurmanji, M., Triantafillou, P., Hayes, J., Triantafillou, E.: Towards unbounded machine unlearning. Advances in Neural Information Processing Systems **36** (2024)
28. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
29. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1765–1773 (2017)
30. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)

31. Mopuri, K.R., Ojha, U., Garg, U., Babu, R.V.: Nag: Network for adversary generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 742–751 (2018)
32. Mopuri, K.R., Uppala, P.K., Babu, R.V.: Ask, acquire, and attack: Data-free uap generation using class impressions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 19–34 (2018)
33. Nguyen, T.T., Huynh, T.T., Nguyen, P.L., Liew, A.W.C., Yin, H., Nguyen, Q.V.H.: A survey of machine unlearning. arXiv preprint arXiv:2209.02299 (2022)
34. Papyan, V., Han, X., Donoho, D.L.: Prevalence of neural collapse during the terminal phase of deep learning training. Proceedings of the National Academy of Sciences **117**(40), 24652–24663 (2020)
35. Ren, Z., Baird, A., Han, J., Zhang, Z., Schuller, B.: Generating and protecting against adversarial attacks for deep speech-based emotion recognition models. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7184–7188. IEEE (2020)
36. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
37. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP). pp. 3–18. IEEE (2017)
38. Song, L., Mittal, P.: Systematic evaluation of privacy risks of machine learning models. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2615–2632 (2021)
39. Tarun, A.K., Chundawat, V.S., Mandal, M., Kankanhalli, M.: Fast yet effective machine unlearning. IEEE Transactions on Neural Networks and Learning Systems (2023)
40. Thudi, A., Deza, G., Chandrasekaran, V., Papernot, N.: Unrolling sgd: Understanding factors influencing machine unlearning. In: 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P). pp. 303–319. IEEE (2022)
41. Thudi, A., Jia, H., Shumailov, I., Papernot, N.: On the necessity of auditable algorithmic definitions for machine unlearning. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 4007–4022 (2022)
42. Tishby, N., Zaslavsky, N.: Deep learning and the information bottleneck principle. In: 2015 ieee information theory workshop (itw). pp. 1–5. IEEE (2015)
43. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. pp. 10347–10357. PMLR (2021)
44. Warnecke, A., Pirch, L., Wressnegger, C., Rieck, K.: Machine unlearning of features and labels. arXiv preprint arXiv:2108.11577 (2021)
45. Ye, Z., Cheng, X., Huang, X.: Fg-uap: Feature-gathering universal adversarial perturbation. In: 2023 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2023)
46. Zhang, C., Benz, P., Imtiaz, T., Kweon, I.S.: Cd-uap: Class discriminative universal adversarial perturbation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 6754–6761 (2020)
47. Zhang, C., Benz, P., Imtiaz, T., Kweon, I.S.: Understanding adversarial examples from the mutual influence of images and perturbations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14521–14530 (2020)