Zero-Shot Machine Unlearning with Proxy Adversarial Data Generation

Huiqiang Chen[1,2 *], **Tianqing Zhu**[1 †], **Xin Yu**[3], **Wanlei Zhou**[1]
[1]City University of Macau, Macau, China
[2]University of Technology Sydney, NSW, Australia
[3]University of Queensland, QLD, Australia
cs.hqchen@gmail.com, {tqzhu, wlzhou}@cityu.edu.mo, xin.yu@uq.edu.au

Abstract

Machine unlearning aims to remove the influence of specific samples from a trained model. A key challenge in this process is over-unlearning, where the model's performance on the remaining data significantly drops due to the change in the model's parameters. Existing unlearning algorithms depend on the remaining data to prevent this issue. As such, these methods are inapplicable in a more practical scenario, where only the unlearning samples are available (i.e., zero-shot unlearning). This paper presents a novel framework, ZS-PAG, to fill this gap. Our approach offers three key innovations: (1) we approximate the inaccessible remaining data by generating adversarial samples; (2) leveraging the generated samples, we pinpoint a specific subspace to perform the unlearning process, therefore preventing over-unlearning in the challenging zero-shot scenario; and (3) we consider the influence of the unlearning process on the remaining samples and design an influence-based pseudo-labeling strategy. As a result, our method further improves the model's performance after unlearning. The proposed method holds a theoretical guarantee, and experiments on various benchmarks validate the effectiveness and superiority of our proposed method over several baselines.

1 Introduction

The success of modern AI systems relies heavily on large-scale datasets [Krizhevsky *et al.*, 2012; Xue *et al.*, 2023; Rajendran *et al.*, 2024]. However, well-trained models are known to memorize their training data [Feldman, 2020], making them vulnerable to privacy attacks wherein adversaries can infer sensitive information by crafting sophisticated queries [Fredrikson *et al.*, 2015; Shokri *et al.*, 2017]. To address these risks, legislatures obligate the model owner to delete users' data upon receiving an unlearning request. Machine unlearning has emerged as a promising paradigm to remove specific data from a well-trained model, effectively erasing their influence as if they were never included in training [Xu *et al.*, 2023].

One challenge of machine unlearning is over-unlearning [Hu *et al.*, 2024]. The model's performance on the remaining data significantly deteriorates after unlearning. Several approaches have been proposed to address this issue. For example, [Foster *et al.*, 2024] suggest only modifying parameters associated with the unlearning samples to prevent affecting the remaining samples. [Wang *et al.*, 2022; Tarun *et al.*, 2023] fine-tune the unlearned model on the remaining data to recover the performance. However, existing methods require accessing the remaining samples, which renders them inapplicable in practice. We envision two critical requirements for a practical unlearning algorithm: **R1** *It should work without accessing the remaining data (i.e., zero-shot)*; **R2** *It should precisely remove only the targeted samples without leading to over-unlearning.* This task is challenging because knowledge is embedded in the model's weights and is highly interconnected [Shwartz-Ziv and Tishby, 2017]. Adjustments to one part of the weights can inadvertently affect other parts. The zero-shot constraint further complicates the challenge.

This paper proposes ZS-PAG (Zero-Shot machine unlearning with Proxy Adversarial data Generation), a novel zero-shot machine unlearning method. In terms of **R1**, we perturb the unlearning samples over the decision boundaries and use the resulting samples to approximate the remaining samples. For **R2**, the unlearning should be more localized to the unlearning samples. Inspired by the phenomena that gradients lie in a subspace [Li *et al.*, 2018], we use the generated adversarial samples to approximate the inaccessible remaining samples and identify a corresponding subspace. The unlearning process is projected into the complementary subspace to prevent over-unlearning. In addition, we propose to optimize the pseudo-labels assigned to the unlearning samples using the influence function [Koh and Liang, 2017]. This ensures that unlearning the pseudo-labeled samples will have a positive influence on the remaining samples. In summary, our main contributions include:

- *Challenging problem*: We consider the zero-shot machine unlearning problem to represent the most practical and challenging setting for machine unlearning.
- *Novel method*: We introduce a novel three-stage zero-

*work done while at University of Queensland
†corresponding author

shot machine unlearning framework, ZS-PAG, effectively preventing over-unlearning in a zero-shot context.

- *Superior performance*: Experiments on various benchmarks demonstrate the efficacy of ZS-PAG. For instance, ZS-PAG outperforms the best baseline method by 6.03% on the CIFAR-100 in a zero-shot setting.

2 Related Works

Machine unlearning. Machine unlearning has received increasing attention over the past years [Xu *et al.*, 2023]. Current research can be broadly categorized into three areas: *Sample-level unlearning*, *feature-level unlearning*, and *class-level unlearning*. *Sample-level unlearning* focuses on the removal of specific training data points [Bourtoule *et al.*, 2021; Cao and Yang, 2015; Izzo *et al.*, 2021; Guo *et al.*, 2020]. This is particularly relevant in scenarios where an individual requests the deletion of their data. *Feature-level unlearning* targets the forgetting of particular features extracted from the training data [Warnecke *et al.*, 2021; Li *et al.*, 2023b]. This is useful when sensitive attributes like race or gender inadvertently influence the model. By unlearning these features, the model can mitigate biases while retaining its overall utility. *Class-level unlearning* concentrates on making the model forget specific classes of data [Chen *et al.*, 2023; Tarun *et al.*, 2023; Golatkar *et al.*, 2020; Fan *et al.*, 2023; Chang *et al.*, 2024]. This is vital when ethical or organizational priorities demand eliminating outdated or harmful classifications. The objective is to remove all knowledge related to that class while maintaining the model's capability to handle other classes effectively.

Mitigating over unlearning. A significant challenge in machine unlearning is over-unlearning [Hu *et al.*, 2024; Shen *et al.*, 2024], where the model's performance significantly degrades after unlearning. Various approaches have been proposed to address this issue. [Chundawat *et al.*, 2023a] employs a teacher-student framework, where the unlearned model is distilled from the original model. During distillation, only knowledge about the remaining samples is retained, while the information related to the unlearning samples is omitted. [Bourtoule *et al.*, 2021] proposes splitting the training data into multiple shards to isolate the influence of specific data points, ensuring their impact does not propagate to the remaining samples. [Foster *et al.*, 2024; Fan *et al.*, 2023] focus on selective model pruning to remove the targeted samples while preserving the model's utility. These methods identify critical parameters essential for general knowledge and selectively prune those linked to the targeted samples, ensuring the model's performance is preserved. [Kurmanji *et al.*, 2024] proposes fine-tuning the original model on both the remaining and unlearning datasets, using gradient negation for the latter to ensure effective unlearning without sacrificing performance.

However, existing methods assume access to the remaining data, which may be invalid. This paper differentiates itself from existing methods by considering unlearning in a zero-shot manner. While some existing works [Chen *et al.*, 2023; Chundawat *et al.*, 2023b; Zhang *et al.*, 2024] also explore zero-shot settings, they face the issue of over-unlearning, resulting in performance degradation on retained samples. In contrast, our method can accurately unlearn the targeted samples without causing over-unlearning. Our work shares a similar intent with [Hoang *et al.*, 2024; Chen *et al.*, 2024; Li *et al.*, 2023a], but several key differences exist. Firstly, the research problems addressed are distinct. This paper focuses on zero-shot unlearning, which presents significant challenges due to the lack of remaining data. Secondly, alongside addressing the issue of over-unlearning, we investigate how unlearning can potentially enhance the model's performance by optimizing the influence.

3 Proposed Method

3.1 Notations

Let $f : \mathbb{R}^d \mapsto \mathbb{R}^C$ be a classifier parameterized by $\theta := \{\mathbf{w}^l\}_{l=1}^L$, and $\mathbf{r}_l^i$ is the input feature w.r.t input x_i at the l-th layer. Define θ_o as an empirical minimizer by fitting f on the training set $\mathcal{D} = \{z_i := (x_i, y_i)\}_{i=1}^N$ with some loss function $\ell(x, y; \theta)$. The training data $\mathcal{D} = \mathcal{D}_u \cup \mathcal{D}_r$ can be partitioned into two disjoint sets of unlearning data $\mathcal{D}_u$ and remaining data $\mathcal{D}_r$. Existing methods assume access to both $\mathcal{D}_r$ and $\mathcal{D}_u$. However, this assumption may be invalid in practice. In this paper, we consider a more stringent setting where we only have access to $f(\theta_o)$ and $\mathcal{D}_u$. Figure 1 provides an overview of the proposed framework.

3.2 Proxy Adversarial Data Generation

In zero-shot machine unlearning, we only have unlearning samples. However, it is hard to prevent over-unlearning without knowledge about the remaining samples. As such, our first step is to generate adversary samples as a proxy for the inaccessible remaining samples.

Starting with an unlearning sample $x_i \in \mathcal{D}_u$, we generate an adversary sample $x_{adv} = x_i + \delta$ via optimizing the additive noise δ. The goal is to deceive the original model into predicting x_{adv} as a pre-specified class $y_{target} \in \bar{c} := [C] \backslash y_i$ in a C classification task. The optimization problem is formulated as follows:

$$\min_{\delta} \mathcal{L}(x_i + \delta, y_{target}; \theta), \quad \text{s.t.} \quad \|\delta\|_p \leq \epsilon. \tag{1}$$

The target label is determined by:

$$y_{target} = \arg\max_{k \in \bar{c}} f_k(x_i; \theta_o), \tag{2}$$

where $f_k(x_i; \theta_o)$ is the k-th component of logits. We use the second high prediction k as the y_{target} for the adversary sample. This choice prioritizes efficiency as perturbing the sample towards this class typically requires less modification compared to forcing a more distant prediction. Essentially, we are perturbing x_i cross decision boundaries. Repeating this process for each unlearning sample, we end with a set of adversary samples residing in regions of the rest classes, which serve as the basis for the subspace estimation in Section 3.3. Directly solving Eq. 1 is prohibitively hard due to high non-convexity. A plethora of algorithms have been proposed to approximate the solution [Zhou *et al.*, 2022]. We use the method proposed by [Madry *et al.*, 2017] in this paper.

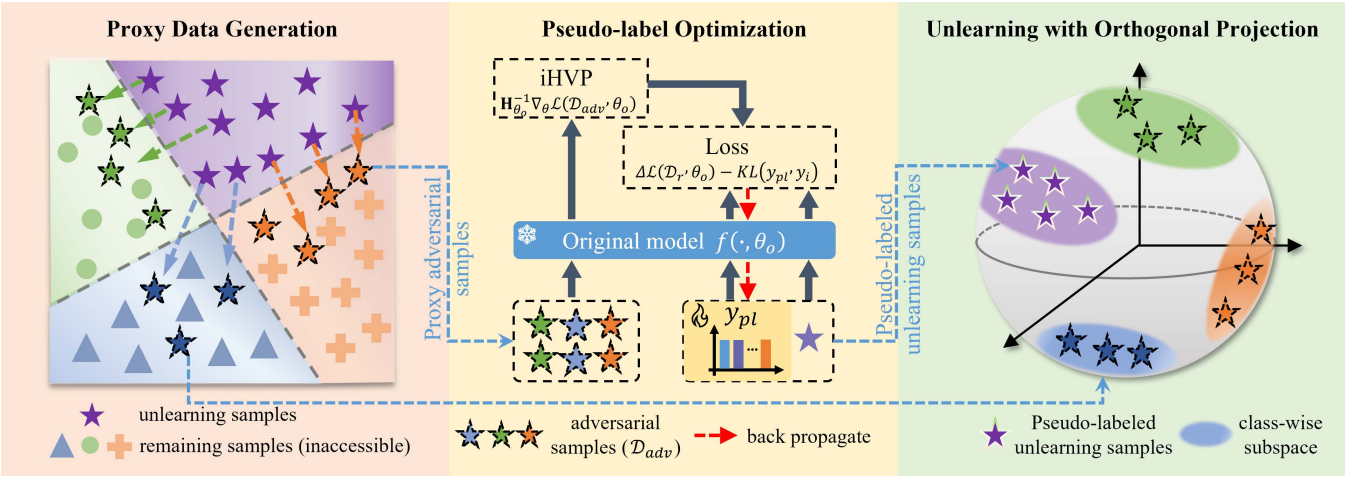


Figure 1: Framework of ZS-PAG. In the first step, we generate adversary samples to approximate the remaining samples. In the second step, we optimize the pseudo-labels assigned to unlearning samples to maximize the positive impact of the unlearning process on the remaining samples. In the third step, we project the unlearning process to a subspace orthogonal to the subspaces of the remaining classes.

3.3 Unlearning with Orthogonal Projection

Existing research reveals that gradients lie in a lower-dimensional subspace spanned by the inputs [Li *et al.*, 2022; Saha *et al.*, 2021]. Inspired by this, we project the unlearning gradients onto a designated subspace where the gradients are irrelevant to the remaining classes. This design effectively prevents over-unlearning.

We identify the subspace of class k by feeding a batch of adversary samples $\mathcal{B}_k = \{(x_i, y_i) \mid (x_i, y_i) \in \mathcal{D}_{adv}, y_i = k\}_{i=1}^{n_{adv}}$ through the original model $f(\theta_o)$. It is expected that using different values of n_{adv} will produce various subspaces and affect the performance of the proposed method. However, as demonstrated in the experiment, setting $n_{adv} = 100$ is sufficient for our needs. At each layer, we compute the subspace corresponding to class k at layer l as follows:

$$\mathbf{U}_l^k \mathbf{\Sigma}_l^k \mathbf{V}_l^{kT} = \text{SVD}(\mathbf{R}_l^k), \tag{3}$$

where $\mathbf{U}_l^k$ contains the basis of the subspace, $\mathbf{R}_l^k = [\mathbf{r}_{l,1}, \mathbf{r}_{l,2}, ..., \mathbf{r}_{l,n_{adv}}]$ represents the input feature maps. The definition of $\mathbf{R}_l^k$ varies by layer type. For fully connected layers, $\mathbf{R}_l^k$ is the raw input feature map. In convolutional layers, $\mathbf{R}_l^k$ is a reshaped input feature map where each row concatenates multiple convolutional patches of the raw input [Liu *et al.*, 2018]. For attention layers consisting of three modules, *i.e.,* ($\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$), we use the raw input of each module as $\mathbf{R}_l^k$ and compute a separate subspace for each module.

To prevent over-unlearning, we project the unlearning gradients onto a subspace orthogonal to the subspaces of the remaining classes. Considering unlearn class c, we iterate over the rest classes $k \in \bar{c}$ to calculate the combined subspace $\mathcal{S}^{\bar{c}}$ as:

$$\mathcal{S}_l^{\bar{c}} = \text{SVD}(\texttt{Concatenate}(\mathcal{S}_l^{\bar{c}}, \mathbf{U}_l^k)), \tag{4}$$

where $\mathcal{S}^{\bar{c}}$ is initialized as an empty set and $\mathcal{S}_l^{\bar{c}}$ is the l-th element. The projection matrix is defined as $\mathbf{P}_l = \mathbf{I} - \mathcal{S}_l^{\bar{c}} \mathcal{S}_l^{\bar{c}T}$. At layer-$l$, the updating rule is:

$$\mathbf{w}_{t+1}^l = \mathbf{w}_t^l - \eta \mathbf{P}_l^{\bar{c}} \nabla_{\mathbf{w}^l} \ell. \tag{5}$$

The following theorem shows unlearning with Eq. 5 prevents over-unlearning.

Theorem 1. *Denote f a neural network parameterized by $\theta := \{\mathbf{w}^l\}_{l=1}^L$, where each $\mathbf{w}^l$ represents the weights of the l-th layer. Assume the loss function $\mathcal{L}_r(\theta) := \frac{1}{|\mathcal{D}_r|} \sum_{\mathcal{D}_r} \ell(f(x_i; \theta), y_i)$ has a Lipschitz continuous gradient with constant L and satisfies the Polyak-Łojasiewicz (PL) inequality [Polyak, 1963] with a constant μ. Under suitable conditions on the learning rate, unlearning with Eq. 5 ensures the unlearned model $f(\theta_u)$ will have approximately the same performance as $f(\theta_o)$, i.e.,*

$$\mathcal{L}_r(\theta_u) \approx \mathcal{L}_r(\theta_o). \tag{6}$$

3.4 Influence-based Pseudo-label Optimization

In addition to preventing over-unlearning, we also propose optimizing the unlearning process's influence on the remaining samples $\mathcal{D}_r$. Denote $\theta^* = \arg\min_\theta \frac{1}{N} \sum_{n=1}^N \ell(z_n; \theta)$ as the empirical risk minimizer. After removing a point z_i from the training set and the optimal parameters became $\theta^*_{-z_i} = \arg\min_\theta \frac{1}{N} \sum_{z_n \neq z_i} \ell(z_n; \theta)$. The influence function gives an efficient approximation to the weight change $\theta^* - \theta^*_{-z_i}$.

Definition 1 (Influence function)**.** *[Koh and Liang, 2017] When upweighting $z_i := (x_i, y_i)$ by ε, we get new parameter as $\theta^*_{\varepsilon, z_i} = \arg\min_\theta \frac{1}{N} \sum_{n=1}^N \ell(z_n; \theta) + \epsilon \ell(z_i; \theta)$. The influence of upweighting z_i on the parameters θ^* is given by*

$$\mathcal{I}_{z_i} = \left. \frac{d\theta^*_{\varepsilon, z_i}}{d\varepsilon} \right|_{\varepsilon=0} = -\mathbf{H}_{\theta^*}^{-1} \nabla_\theta \ell(z_i; \theta^*), \tag{7}$$

where $\mathbf{H}_{\theta^*} = \frac{1}{N} \sum_{n=1}^N \nabla_\theta^2 \ell(z_n; \theta^*)$ is the Hessian of the loss function on the training data.

Assume we have access to $\mathcal{D}_r$ for now. Consider upweighting a pseudo-labeled unlearning sample $z_i := (x_i, y_{pl})$ by ε.

The change in loss of the remaining sample $x \in \mathcal{D}_r$ is estimated as:

$$\begin{aligned}\Delta\mathcal{L}(\mathcal{D}_r, \theta_u) &= \mathcal{L}(\mathcal{D}_r, \theta_u) - \mathcal{L}(\mathcal{D}_r, \theta_o) \\ &\approx \nabla_\theta \mathcal{L}(\mathcal{D}_r, \theta_o)^T (\theta_u - \theta_o) \\ &\approx \nabla_\theta \mathcal{L}(\mathcal{D}_r, \theta_o)^T \varepsilon \mathcal{I}_{z_i},\end{aligned} \tag{8}$$

where the last step is the application of influence function $\mathcal{I}_{z_i}$. Replace $\mathcal{I}_{x_i}$ with Eq. 7, we get:

$$\Delta\mathcal{L}(\mathcal{D}_r, \theta_u) \approx -\varepsilon \nabla_\theta \mathcal{L}(\mathcal{D}_r, \theta_o)^T \mathbf{H}_{\theta_o}^{-1} \nabla_\theta \ell(x_i, y_{pl}; \theta_o). \tag{9}$$

Removing a sample (x_i, y_{pl}) is equivalent to setting $\varepsilon = -\frac{1}{N}$. Therefore, we approximate the change of loss on $\mathcal{D}_r$ after unlearning x_i as:

$$\Delta\mathcal{L}(\mathcal{D}_r, \theta_u) \approx \frac{1}{N} \nabla_\theta \mathcal{L}(\mathcal{D}_r, \theta_o)^T \mathbf{H}_{\theta_o}^{-1} \nabla_\theta \ell(x_i, y_{pl}; \theta_o). \tag{10}$$

Calculating $\mathbf{H}_{\theta_o}^{-1}$ is costly for DNNs. Therefore, iterative methods [Sattigeri *et al.*, 2022] are often used in practice to approximate $\mathbf{H}_{\theta_o}^{-1} \nabla_\theta \mathcal{L}(\mathcal{D}_r, \theta_o)$.

We random initialize $y_{pl} \in \mathbb{R}^C$ and optimize it to decrease $\Delta\mathcal{L}(\mathcal{D}_r, \theta_u)$ approximated by Eq. 10. However, minimizing $\Delta\mathcal{L}(\mathcal{D}_r, \theta_u)$ alone may lead y_{pl} collapse into the ground-truth label y_i. As such, we add a regularization term to penalize the similarity between y_{pl} and y_i. This could be done via any similarity metric. Our experiment empirically finds that the Kullback-Leibler (KL) divergence suffices our needs. By optimizing y_{pl}, we ensure the model will benefit from the unlearning process. This results in improved performance after unlearning. In a zero-shot unlearning setting, we approximate $\mathcal{D}_r$ with $\mathcal{D}_{adv}$ generated as described in Section 3.2. Our empirical results demonstrate that $\mathcal{D}_{adv}$ serves as an adequate proxy for $\mathcal{D}_r$.

4 Experiment

4.1 Experimental Setup

Dataset and model architecture. Following previous works, we evaluate the proposed method on four benchmarks: Facescrub [Ng and Winkler, 2014], SVHN [Netzer *et al.*, 2011], CIFAR-10 and CIFAR-100 [Krizhevsky *et al.*, 2009]. We apply four representative network architectures in our experiments: AlexNet [Krizhevsky *et al.*, 2012], VGG [Simonyan and Zisserman, 2014], ResNet [He *et al.*, 2015], and ViT [Dosovitskiy *et al.*, 2020].

Baselines. We compare our approach with several baselines. The first five methods assume access to $\mathcal{D}_r$: (1) FT [Warnecke *et al.*, 2021], (2) Neggrad [Kurmanji *et al.*, 2024], (3) BadT [Chundawat *et al.*, 2023a], (4) SalUn [Fan *et al.*, 2023], and (5) Fisher [Golatkar *et al.*, 2020]. To make a fair comparison, we included the results of their zero-shot versions. We also add two baselines that do not use $\mathcal{D}_r$: (6) BU [Chen *et al.*, 2023] and (7) GKT [Chundawat *et al.*, 2023b]. Finally, we include the results of (8) Retrain.

Evaluation metrics and implement details. Following the literature, we assess the unlearned model with three metrics: 1) Acc_{ut}: Accuracy on the testing set of unlearning classes. In a class unlearning setting, the unlearned model should have zero accuracy on the unlearning classes, matching a retrained model; 2) Acc_{mia}: Accuracy of membership inference attack (MIA). We train an attack model to predict the membership of unlearning samples in the training set. As noted by [Fan *et al.*, 2023; Chen *et al.*, 2023], the closer this metric is to that of the retrained model, the better the performance of the unlearning algorithm; 3) Acc_{rt}: Accuracy on testing set of remaining classes. We measure the degree of over-unlearning. An idea unlearned model should not decrease Acc_{rt}. All results are averaged over three different runs. We utilize projected gradient descent [Madry *et al.*, 2017] to generate adversary samples $\mathcal{D}_{adv}$ in the experiment. Note our method is compatible with any adversary attack strategy.

4.2 Single-class Unlearning

We randomly select one class for SVHN and CIFAR10, two classes for Facescrub, and ten classes for CIFAR100 as the unlearning class(es) and compare our method with the baselines. Results in Table 1 reveal the following key findings:

1) *Existing unlearning methods rely on $\mathcal{D}_r$ to prevent over-unlearning.* There exists a significant gap in $Acc_{\mathcal{D}_{ut}}$ between these methods and their zero-shot versions. For instance, on the SVHN dataset, the FT method fine-tunes the original model using a combination of $\mathcal{D}_r$ and random labeled samples from $\mathcal{D}_u$. As a result, $Acc_{\mathcal{D}_{ut}}$ of the unlearned model is decreased to 0.95%, while $Acc_{\mathcal{D}_{rt}}$ remains at 95.97%, which is comparable to the original model. However the $Acc_{\mathcal{D}_{rt}}$ of FT-zs drops by more than 15.5%. This comparison highlights FT's reliance on $\mathcal{D}_r$ to prevent over-unlearning. This conclusion also applies to Neggrad, BadT, and SalUN.

2) *ZS-PAG out-stands in zero-shot unlearning*. For example, on the CIFAR-10 dataset, $Acc_{\mathcal{D}_{ut}}$ and $Acc_{\mathcal{D}_{rt}}$ of ZS-PAG are 1.40% and 85.47%, respectively. In comparison, the original model yields 63.97% and 83.80%, respectively. ZS-PAG effectively removed the information of the unlearning classes from the model and improved the model's performance in the remaining class. The improvement in $Acc_{\mathcal{D}_{rt}}$ can be attributed to the optimized pseudo-labels in ZS-PAG, which maximize the influence of the unlearning process on the remaining samples. Notably, ZS-PAG improves $Acc_{\mathcal{D}_{rt}}$ by 1.53%, 1.67%, and 2.23% on the FashionMNIST, CIFAR-10, and CIFAR-100 dataset, respectively.

4.3 Multi-class Unlearning

We extend the comparison to multi-classes unlearning by randomly unlearning 3 classes from the SVHN dataset and comparing ZS-PAG against baselines. As shown in Table 2, the baseline methods exhibit a similar performance drop in $Acc_{\mathcal{D}rt}$ as seen in Table 1. What's worse, the over-unlearning issue worsens when unlearning more classes. For instance, the gap of $Acc_{\mathcal{D}_{rt}}$ between BadT and BadT-zs is 17.79% in Table 1 for single class unlearning. And this gap increases to 74.48% when unlearning 3 classes. In contrast, ZS-PAG successfully prevents over-unlearning and demonstrates comparable even higher $Acc_{\mathcal{D}_{rt}}$ than baselines, despite these methods using remaining samples during unlearning.

Comparing ZS-PAG to the original model across Tables 1 and 2, we observe increasing gains in $Acc_{\mathcal{D}_{rt}}$ as more classes are unlearned: 0.11% for 1 class and 1.19% for 3classes. We

Approach	Facescrub/ResNet		SVHN/VGG		CIFAR-10/ViT		CIFAR-100/ResNet	
	$Acc_{\mathcal{D}_{rt}}(\uparrow)$	$Acc_{\mathcal{D}_{ut}}(\downarrow)$	$Acc_{\mathcal{D}_{rt}}(\uparrow)$	$Acc_{\mathcal{D}_{ut}}(\downarrow)$	$Acc_{\mathcal{D}_{rt}}(\uparrow)$	$Acc_{\mathcal{D}_{ut}}(\downarrow)$	$Acc_{\mathcal{D}_{rt}}(\uparrow)$	$Acc_{\mathcal{D}_{ut}}(\downarrow)$
Original	$96.17_{\pm 0.05}$	$96.05_{\pm 3.20}$	$95.52_{\pm 0.12}$	$91.30_{\pm 0.30}$	$83.80_{\pm 1.16}$	$63.97_{\pm 0.46}$	$73.31_{\pm 0.31}$	$72.07_{\pm 1.18}$
Retrain	$96.31_{\pm 0.15}$	$0.00_{\pm 0.00}$	$95.56_{\pm 0.23}$	$0.00_{\pm 0.00}$	$86.57_{\pm 0.28}$	$0.00_{\pm 0.00}$	$75.36_{\pm 0.34}$	$0.00_{\pm 0.00}$
FT	$74.69_{\pm 6.31}$	$0.00_{\pm 0.00}$	$95.97_{\pm 0.05}$	$0.95_{\pm 0.51}$	$87.60_{\pm 0.65}$	$6.43_{\pm 1.68}$	$74.68_{\pm 0.03}$	$2.33_{\pm 0.94}$
FT-zs	$78.72_{\pm 0.25}$	$0.00_{\pm 3.34}$	$80.41_{\pm 6.60}$	$5.91_{\pm 2.80}$	$80.69_{\pm 1.64}$	$4.70_{\pm 1.55}$	$66.63_{\pm 0.22}$	$0.33_{\pm 0.47}$
Neggrad	$65.59_{\pm 2.10}$	$0.00_{\pm 0.00}$	$96.15_{\pm 0.08}$	$0.02_{\pm 0.03}$	$88.29_{\pm 0.63}$	$4.10_{\pm 2.05}$	$75.15_{\pm 0.15}$	$3.00_{\pm 1.63}$
Neggrad-zs	$79.09_{\pm 1.05}$	$3.85_{\pm 0.33}$	$57.02_{\pm 34.14}$	$9.40_{\pm 13.30}$	$76.89_{\pm 7.71}$	$1.77_{\pm 1.53}$	$66.01_{\pm 4.88}$	$2.33_{\pm 3.30}$
BadT	$87.51_{\pm 1.03}$	$6.41_{\pm 4.80}$	$96.01_{\pm 0.06}$	$3.27_{\pm 0.65}$	$85.51_{\pm 1.29}$	$2.20_{\pm 1.82}$	$71.94_{\pm 0.32}$	$8.33_{\pm 2.49}$
BadT-zs	$74.60_{\pm 1.60}$	$5.77_{\pm 2.72}$	$78.22_{\pm 4.63}$	$2.30_{\pm 0.67}$	$68.54_{\pm 4.58}$	$4.23_{\pm 2.03}$	$24.69_{\pm 16.71}$	$8.67_{\pm 6.34}$
SalUn	$85.20_{\pm 1.63}$	$1.28_{\pm 1.81}$	$95.85_{\pm 0.06}$	$3.48_{\pm 1.14}$	$87.51_{\pm 0.75}$	$6.47_{\pm 1.62}$	$74.68_{\pm 0.10}$	$1.67_{\pm 1.25}$
SalUn-zs	$77.50_{\pm 0.63}$	$0.00_{\pm 0.00}$	$80.38_{\pm 6.62}$	$5.86_{\pm 2.66}$	$80.77_{\pm 1.81}$	$4.60_{\pm 1.39}$	$67.82_{\pm 1.41}$	$0.00_{\pm 0.00}$
Fisher	$51.88_{\pm 1.36}$	$0.00_{\pm 0.00}$	$92.69_{\pm 1.03}$	$0.54_{\pm 0.67}$	$84.51_{\pm 1.35}$	$6.27_{\pm 7.1}$	$50.03_{\pm 1.93}$	$0.00_{\pm 0.00}$
BU	$77.21_{\pm 1.98}$	$0.64_{\pm 0.91}$	$82.19_{\pm 6.52}$	$9.26_{\pm 4.06}$	$80.78_{\pm 1.72}$	$4.67_{\pm 1.55}$	$69.51_{\pm 0.39}$	$3.00_{\pm 0.82}$
GKT	$73.10_{\pm 2.52}$	$0.99_{\pm 0.17}$	$94.25_{\pm 0.79}$	$2.95_{\pm 2.14}$	$74.67_{\pm 6.35}$	$9.83_{\pm 5.97}$	$63.53_{\pm 0.78}$	$6.00_{\pm 6.48}$
ZS-PAG	$96.48_{\pm 0.22}$	$1.92_{\pm 0.21}$	$95.63_{\pm 0.07}$	$0.21_{\pm 0.11}$	$85.47_{\pm 1.11}$	$1.40_{\pm 0.14}$	$75.54_{\pm 0.33}$	$2.00_{\pm 1.63}$

Table 1: **Single-class unlearning.** Comparison of ZS-PAG and baselines across various datasets and model architectures. Results highlighted in light gray are obtained in a zero-shot setting.

Approach	$Acc_{\mathcal{D}_{rt}}(\uparrow)$	$Acc_{\mathcal{D}_{ut}}(\downarrow)$	Acc_{mia}
Original	$95.39_{\pm 0.15}$	$94.41_{\pm 0.13}$	$67.57_{\pm 0.55}$
Retrain	$95.90_{\pm 1.15}$	$0.00_{\pm 0.00}$	$39.38_{\pm 4.86}$
FT	$96.67_{\pm 0.07}$	$0.03_{\pm 0.01}$	$36.28_{\pm 4.15}$
FT-zs	$52.38_{\pm 6.13}$	$0.23_{\pm 0.12}$	$51.30_{\pm 2.61}$
Neggrad	$95.06_{\pm 2.74}$	$0.00_{\pm 0.00}$	$52.44_{\pm 0.90}$
Neggrad-zs	$32.43_{\pm 16.99}$	$1.24_{\pm 1.75}$	$49.73_{\pm 1.11}$
BadT	$92.96_{\pm 0.55}$	$0.61_{\pm 0.87}$	$64.00_{\pm 5.13}$
BadT-zs	$18.48_{\pm 4.61}$	$0.25_{\pm 0.35}$	$40.61_{\pm 3.78}$
SalUn	$96.48_{\pm 0.12}$	$0.14_{\pm 0.02}$	$40.82_{\pm 4.20}$
SalUn-zs	$52.48_{\pm 6.14}$	$0.24_{\pm 0.14}$	$49.91_{\pm 2.24}$
Fisher	$95.18_{\pm 0.07}$	$12.36_{\pm 10.91}$	$50.06_{\pm 8.57}$
BU	$53.64_{\pm 5.75}$	$0.45_{\pm 0.23}$	$47.95_{\pm 0.90}$
GKT	$72.78_{\pm 3.14}$	$0.00_{\pm 0.00}$	$43.19_{\pm 0.76}$
ZS-PAG	$96.58_{\pm 0.18}$	$0.31_{\pm 0.14}$	$47.81_{\pm 3.60}$

Table 2: **Multi-class unlearning**. We randomly unlearn 3 classes from SVHN and compare our method against baselines. Results highlighted in light gray are obtained in a zero-shot setting.

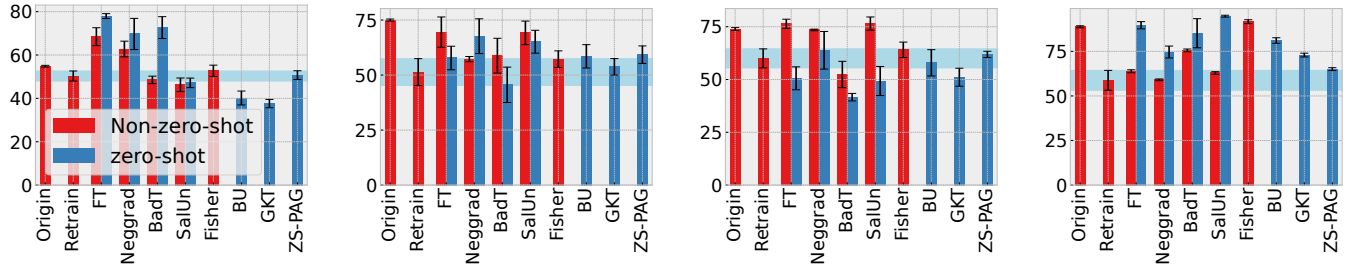
accumulate a greater impact from each unlearning sample on the remaining samples when unlearning more classes, resulting in an improved overall outcome.

4.4 Unlearning Guarantee

MIA. Following the prior arts [Fan *et al.*, 2023; Chen *et al.*, 2023], we use the same setting as Sec. 4.2 and perform MIA against the unlearned model to probe the retained information about the unlearning classes in the unlearned model. Results are presented in Figure 2. Note a large deviation of Acc_{mia} from the retrained model may leak information about the unlearning samples. Therefore, Acc_{mia} values closer to the retrained model are more desirable. The optimal regions are highlighted in Figure 2. The baselines exhibit Acc_{mia} values either significantly higher or lower than that of the retrained model. For example, on the SVHN dataset in Figure 2b, FT achieves the highest Acc_{mia} among all unlearning methods. This indicates an unsuccessful unlearning. In contrast, the results of ZS-PAG lie close to the optimal region on all datasets, suggesting a successful unlearning.

Visualize attention map. We plot the GradCAM [Selvaraju *et al.*, 2017] results on samples from the Facescrub dataset to evaluate the effectiveness of ZS-PAG. Two classes were unlearned from the original model to obtain the unlearned model. The results in Figure 3 demonstrate the effectiveness of our unlearning approach in selectively erasing target samples while preserving the model's performance on the remaining data. The first row shows heatmaps for the original model, and the second row corresponds to the unlearned model. For unlearning samples, the original model focuses on the eye regions, which are critical for identity recognition, whereas the unlearned model displays random attention, indicating successful forgetting. For the remaining samples, both models exhibit consistent attention to the eye regions, demonstrating that ZS-PAG preserves the model's functionality on non-target data.

Backdoor attack-based metric. Beyond MIA and attention maps, we additionally employ a backdoor attack-based metric to further evaluate the unlearning effectiveness of ZS-PAG and its robustness against over-unlearning, following the protocol in [Chundawat *et al.*, 2023b]. Specifically, we train a ViT model on the CIFAR-10 dataset, implant a backdoor by adding visual triggers to samples from class 1, and relabel them as class 8 to simulate a targeted attack scenario. We apply ZS-PAG to unlearn class 1 and remove its influence. The results, shown in Tab. 3, demonstrate that the attack accuracy (Acc_{attack}) drops dramatically from 89.09 to 0.24 after unlearning. This significant reduction indicates the successful elimination of the backdoor functionality and further confirms the effectiveness of ZS-PAG in erasing sensitive information while preserving the integrity of the remaining model.



(a) Facescrub (b) SVHN (c) CIFAR-10 (d) CIFAR-100

Figure 2: MIA results of ZS-PAG and baselines for single-class unlearning. Results outside the highlighted optimal region may leak information about the unlearning samples.

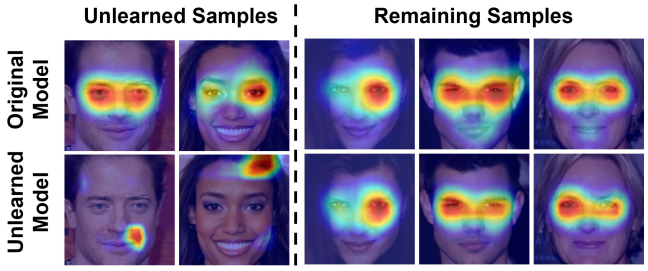


Figure 3: GradCAM results on unlearned and remaining samples for the original and unlearned models obtained by ZS-PAG.

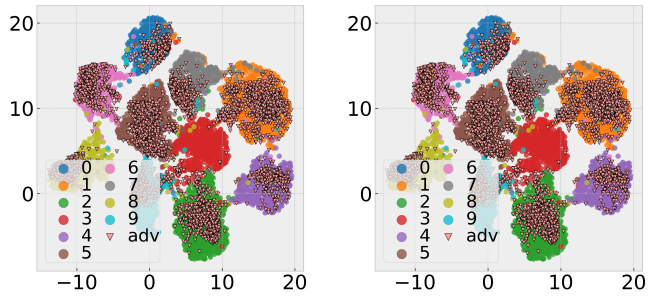
CIFAR10/ViT	Original	Retrain	ZS-PAG
$Acc_{D_{rt}}(\uparrow)$	84.12	83.96	85.33
$Acc_{D_{ut}}(\downarrow)$	67.20	0.00	0.96
$Acc_{attack}(\downarrow)$	89.09	–	0.24

Table 3: **Backdoor attack-based metric.** We unlearn 1 class from a backdoored ViT.

4.5 Ablation Studies

In this section, we analyze key components of the proposed ZS-PAG framework. These studies focus on (1) the distribution of adversarial samples, (2) the influence of adversarial attack, and (3) the contributions of subspace projection and pseudo-labeling. The results provide insights into the robustness and effectiveness of the proposed method.

Distribution of adversarial samples. To check how well the generated adversarial samples represent the remaining data, we visualize their distribution in relation to real samples. Using the SVHN and CIFAR-10 datasets with class 3 designated as the unlearning class, we generate adversarial samples to approximate the remaining classes. Figure 4 demonstrates that the adversarial samples closely align with real data distributions, particularly for classes adjacent to the unlearning class. In both datasets, we set class 3 as the unlearning class. The generated samples closely overlap with the real samples, demonstrating a good approximation. This proximity arises from the reduced perturbation required to shift samples to nearby decision boundaries. In comparison, distant classes require larger perturbations. Nevertheless, as shown in the following, a limited number of adversarial samples per class suffices to estimate class-wise subspaces effectively.

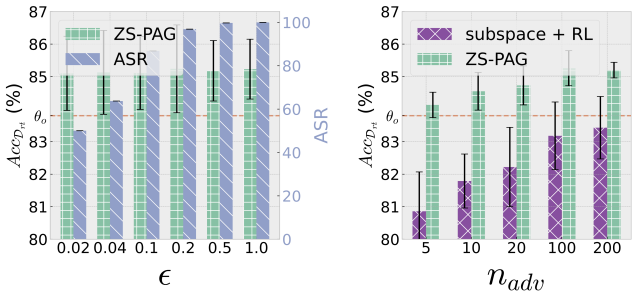


(a) SVHN (b) CIFAR-10

Figure 4: Distribution of generated adversarial samples. We set class 3 as the unlearning class and generate adversarial samples to approximate samples of the rest of the classes.

Influence of adversary attack. We estimate the inaccessible remaining samples using generated adversary samples in our method. A pertinent question arises: *How does the adversary attack affect the effectiveness of our method?* We conduct ablation studies using the same settings as in Section 4.2. These studies address this question by examining two aspects: 1) the influence of attack success rate (ASR), defined as the ratio of samples successfully identified as other classes; and 2) the choice of different attack methods.

Influence of ASR: We generate adversarial samples with varying noise bound ε. As shown in Figure 5a, increasing ε results in higher ASR due to larger perturbations. Notably, $Acc_{\mathcal{D}_{rt}}$ remains stable around 85%, indicating that ZS-PAG is robust to variations in ASR. Additional experiments, as depicted in Figure 5b, confirm that as the number of generated adversarial samples per class n_{adv} increases, the gap between the performance of the subspace-only method (*Subspace + RL*) and the original model narrows, demonstrating that a sufficient number of adversarial samples enhances subspace estimation and mitigates over-unlearning. This observation aligns with the results shown in Figure 5a, where we observe consistent performance across ASR levels ranging from 52% to 99%. In the context of class unlearning, a substantial number of samples are available for adversarial attacks. Consequently, even a low ASR provides enough adversarial sam-



(a) (b)

Figure 5: Influence of adversary attack on CIFAR-10 dataset for single-class unlearning. (a) We use various levels of noise bound ε when finding adversary samples. (b) We estimate the null space with different numbers of adversary samples.

CIFAR-10/ViT	$Acc_{\mathcal{D}_{rt}}(\uparrow)$	$Acc_{\mathcal{D}_{ut}}(\downarrow)$
Original	$83.80_{\pm 1.16}$	$63.97_{\pm 0.46}$
Retrain	$86.57_{\pm 0.28}$	$0.00_{\pm 0.00}$
PGD	$85.47_{\pm 1.49}$	$1.40_{\pm 0.14}$
FGSM	$85.43_{\pm 1.06}$	$2.00_{\pm 0.16}$
CW	$84.46_{\pm 1.11}$	$0.90_{\pm 0.08}$
DeepFool	$85.49_{\pm 0.95}$	$0.720_{\pm 0.11}$

Table 4: **Influence of attack method.** We evaluate the impact of different adversarial attack methods on the unlearning performance.

ples to meet our requirements.

Influence of different choices of attack methods: We compare four different attack methods, 1) PGD [Madry *et al.*, 2017] (the default method in this paper), 2) FGSM [Goodfellow *et al.*, 2015], 3) CW [Carlini and Wagner, 2017], and 4) DeepFool [Moosavi-Dezfooli *et al.*, 2016]. Results in Table 4 show that all methods achieve comparable performance in terms of $Acc_{\mathcal{D}_{rt}}$, ranging from 84.46% to 85.49%. Minor differences in $Acc_{\mathcal{D}_{ut}}$ are observed. These results demonstrate that ZS-PAG is robust to the choice of attack method. This robustness stems from the fact that ZS-PAG primarily focuses on probing the decision boundaries of the model rather than requiring high-quality adversarial samples. By focusing on the general properties of decision boundaries, our method accurately identifies the subspaces needed for unlearning, regardless of the attack method used.

Contribution of subspace projection. The efficacy of subspace projection in mitigating over-unlearning is analyzed through three experiments: (1) vanilla Random Labeling (*RL*), (2) RL with a randomly generated subspace (*Random Subspace+RL*), and (3) RL with the estimated subspace derived from adversarial samples (*Estimate Subspace+RL*). As shown in Table 5, *RL* reduces remaining class accuracy $Acc_{\mathcal{D}_{rt}}$ from from 95.52% to 80.41%, causing significant over-unlearning. Employing a random subspace exacerbates this issue, lowering $Acc_{\mathcal{D}_{rt}}$ to 59.08%. In contrast, the estimated subspace effectively prevents over-unlearning, maintaining $Acc_{\mathcal{D}_{rt}}$ at 95.01%. These results suggest that the unlearning gradient must be approached with caution. Mitigating over-unlearning requires identifying a subspace significant to the remaining samples..

SVHN/VGG	$Acc_{\mathcal{D}_{rt}}(\uparrow)$	$Acc_{\mathcal{D}_{ut}}(\downarrow)$
Original	$95.52_{\pm 0.12}$	$91.30_{\pm 0.30}$
Retrain	$95.56_{\pm 0.23}$	$0.00_{\pm 0.00}$
RL	$80.41_{\pm 6.60}$	$5.91_{\pm 2.80}$
Random Subspace + *RL*	$59.08_{\pm 4.71}$	$1.15_{\pm 1.05}$
Estimate Subspace + *RL*	$95.01_{\pm 0.33}$	$1.46_{\pm 1.14}$

Table 5: **Contribution of Subspace Projection**. We conduct an ablation study to evaluate the contribution of subspace projection.

Contribution of pseudo-labeling. We compare ZS-PAG with a subspace-only method that excludes pseudo-labeling. As illustrated in Figure 5b, the pseudo-labeling strategy consistently improves remaining class accuracy $Acc_{\mathcal{D}_{rt}}$, with performance gains ranging from 1.77% to 3.27%, depending on the number of adversarial samples per class n_{adv}. This improvement underscores the effectiveness of influence-based pseudo-label optimization in enhancing the model's overall performance.

Computational cost of ZS-PAG. We conduct the experiment on an NVIDIA RTX 4090 GPU. We fix the unlearning epochs to 10 for a fair comparison. Tab. 6 indicates the running time and unlearning performance. ZS-PAG uses significantly less running time than Retrain and achieves the best unlearning performance, with the highest $Acc_{D_{rt}}$. Compared to the second-best method, Neggrad, ZS-PAG incurs less computing overhead and achieves better unlearning performance.

	Retrain	SalUn	FT	BU	Neggrad	Fisher	ZS-PAG
Time	1h22m7s	5m13s	5m14s	5m16s	10m12s	21m34s	8m28s
$Acc_{D_{rt}}$	75.36	74.68	74.68	69.51	75.15	50.03	**75.54**
$Acc_{D_{ut}}$	0.00	1.67	2.33	3.00	3.00	0.00	2.00

Table 6: **Computation overhead**.We unlearn 1 class from CIFAR100/ResNet18.

5 Conclusion

This paper presents ZS-PAG, a novel approach to zero-shot machine unlearning. ZS-PAG approximates the inaccessible remaining samples with adversary samples and confines the unlearning process within a specified subspace. This effectively prevents over-unlearning. Additionally, ZS-PAG integrates influence-based optimization techniques to enhance the unlearned model's performance further. Our theoretical analysis provides robust support for this approach, and empirical results convincingly demonstrate the superior performance of ZS-PAG over existing methods. Moreover, our method exhibits robustness across various model architectures, underscoring its effectiveness and adaptability.

Acknowledgments

This research is partially supported by NSFC-FDCT under its Joint Scientific Research Project Fund (Grant No. 0051/2022/AFJ)

References

[Bourtoule *et al.*, 2021] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[Cao and Yang, 2015] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017.

[Chang *et al.*, 2024] Wenhan Chang, Tianqing Zhu, Heng Xu, Wenjian Liu, and Wanlei Zhou. Class machine unlearning for complex data via concepts inference and data poisoning, 2024.

[Chen *et al.*, 2023] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7766–7775, 2023.

[Chen *et al.*, 2024] Huiqiang Chen, Tianqing Zhu, Xin Yu, and Wanlei Zhou. Machine unlearning via null space calibration. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 358–366. International Joint Conferences on Artificial Intelligence Organization, 8 2024. Main Track.

[Chundawat *et al.*, 2023a] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7210–7217, 2023.

[Chundawat *et al.*, 2023b] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2023.

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

[Fan *et al.*, 2023] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, 2023.

[Feldman, 2020] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

[Foster *et al.*, 2024] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12043–12051, 2024.

[Fredrikson *et al.*, 2015] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

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

[Goodfellow *et al.*, 2015] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[Guo *et al.*, 2020] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning*, pages 3832–3842. PMLR, 2020.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. arxiv e-prints. *arXiv preprint arXiv:1512.03385*, 10, 2015.

[Hoang *et al.*, 2024] Tuan Hoang, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Learn to unlearn for deep neural networks: Minimizing unlearning interference with gradient projection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4819–4828, January 2024.

[Hu *et al.*, 2024] Hongsheng Hu, Shuo Wang, Jiamin Chang, Haonan Zhong, Ruoxi Sun, Shuang Hao, Haojin Zhu, and Minhui Xue. A duty to forget, a right to be assured? exposing vulnerabilities in machine unlearning services, 2024.

[Izzo *et al.*, 2021] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models, 2021.

[Koh and Liang, 2017] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

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

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[Kurmanji *et al.*, 2024] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards un-

bounded machine unlearning. *Advances in neural information processing systems*, 36, 2024.

[Li *et al.*, 2018] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

[Li *et al.*, 2022] Tao Li, Lei Tan, Zhehao Huang, Qinghua Tao, Yipeng Liu, and Xiaolin Huang. Low dimensional trajectory hypothesis is true: Dnns can be trained in tiny subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3411–3420, 2022.

[Li *et al.*, 2023a] Guanghao Li, Li Shen, Yan Sun, Yue Hu, Han Hu, and Dacheng Tao. Subspace based federated unlearning, 2023.

[Li *et al.*, 2023b] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Yizhao Zhang, Zhongxuan Han, Dan Meng, and Jun Wang. Making users indistinguishable: Attribute-wise unlearning in recommender systems. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 984–994, 2023.

[Liu *et al.*, 2018] Zhenhua Liu, Jizheng Xu, Xiulian Peng, and Ruiqin Xiong. Frequency-domain dynamic pruning for convolutional neural networks. *Advances in neural information processing systems*, 31, 2018.

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

[Moosavi-Dezfooli *et al.*, 2016] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2016.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011.

[Ng and Winkler, 2014] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *2014 IEEE international conference on image processing (ICIP)*, pages 343–347. IEEE, 2014.

[Polyak, 1963] B.T. Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

[Rajendran *et al.*, 2024] Megani Rajendran, Chek Tien Tan, Indriyati Atmosukarto, Aik Beng Ng, and Simon See. Review on synergizing the metaverse and ai-driven synthetic data: enhancing virtual realms and activity recognition in computer vision. *Visual Intelligence*, 2(1):27, 2024.

[Saha *et al.*, 2021] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021.

[Sattigeri *et al.*, 2022] Prasanna Sattigeri, Soumya Ghosh, Inkit Padhi, Pierre Dognin, and Kush R Varshney. Fair infinitesimal jackknife: Mitigating the influence of biased training data points without refitting. *Advances in Neural Information Processing Systems*, 35:35894–35906, 2022.

[Selvaraju *et al.*, 2017] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[Shen *et al.*, 2024] Shaofei Shen, Chenhao Zhang, Alina Bialkowski, Weitong Chen, and Miao Xu. Camu: Disentangling causal effects in deep model unlearning. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 779–787. SIAM, 2024.

[Shokri *et al.*, 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models, 2017.

[Shwartz-Ziv and Tishby, 2017] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

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

[Tarun *et al.*, 2023] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[Wang *et al.*, 2022] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM Web Conference 2022*, pages 622–632, 2022.

[Warnecke *et al.*, 2021] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*, 2021.

[Xu *et al.*, 2023] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S Yu. Machine unlearning: A survey. *ACM Computing Surveys*, 56(1):1–36, 2023.

[Xue *et al.*, 2023] Wanli Xue, Jingze Liu, Siyi Yan, Yuxi Zhou, Tiantian Yuan, and Qing Guo. Alleviating data insufficiency for chinese sign language recognition. *Visual Intelligence*, 1(1):26, 2023.

[Zhang *et al.*, 2024] Chenhao Zhang, Shaofei Shen, Weitong Chen, and Miao Xu. Toward efficient data-free unlearning, 2024.

[Zhou *et al.*, 2022] Shuai Zhou, Chi Liu, Dayong Ye, Tianqing Zhu, Wanlei Zhou, and Philip S Yu. Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. *ACM Computing Surveys*, 55(8):1–39, 2022.