Fine-Grained and Efficient Self-Unlearning with Layered Iteration

Hongyi Lyu[1], **Xuyun Zhang**[1*], **Hongsheng Hu**[2*], **Shuo Wang**[3], **Chaoxiang He**[3], **Lianyong Qi**[4]
[1]Macquarie University
[2]University of Newcastle
[3]Shanghai Jiao Tong University
[4]China University of Petroleum (East China)

Abstract

As machine learning models become widely deployed in data-driven applications, ensuring compliance with the *right to be forgotten* as required by many privacy regulations is vital for safeguarding user privacy. To forget the given data, existing re-labeling based unlearning methods employ a single-step adjustment scheme that revises the decision boundaries in one re-labeling phase. However, such single-step approaches lead to coarse-grained changes in decision boundaries among the remaining classes and impose adverse effects on the model utility. To address these limitations, we propose *Self-Unlearning with Layered Iteration (SULI)*, a novel unlearning approach that introduces a layered iteration strategy to re-label the forgetting data iteratively and refine the decision boundaries progressively. We further develop a *Selective Probability Adjustment (SPA)* technique, which uses a soft-label mechanism to promote smoother decision-boundary transitions. Comprehensive experiments on three benchmark datasets demonstrate that SULI achieves superior performance in effectiveness, efficiency, and privacy compared to the state-of-the-art baselines in both class-wise and instance-wise unlearning scenarios. The source code is released at https://github.com/Hongyi-Lyu-MQ/SULI.

1 Introduction

The rapid growth of machine learning (ML) applications has revolutionized operational efficiency and user experiences and introduced critical challenges in data privacy and regulatory compliance. The General Data Protection Regulation (GDPR) [Voigt and Von dem Bussche, 2017] emphasizes the *Right to be Forgotten*, i.e., organizations must remove personal data upon request both precisely and efficiently. This requirement places considerable challenges on data handlers in ensuring effective compliance with privacy protection.

Machine unlearning has emerged as a vital attempt to meet these regulatory demands while mitigating privacy threats such as data extraction attacks [Carlini *et al.*, 2021] and membership inference attacks [Shokri *et al.*, 2017; Hu *et al.*, 2022]. It is still a challenge to efficiently remove the contribution of specific training data from an ML model while maintaining satisfactory model utility. Traditional approaches typically retrain models from scratch using the retaining data only [Nguyen *et al.*, 2022], which achieves perfect forgetting, but such methods are often computationally expensive and impractical in most real-world scenarios [Foster *et al.*, 2024]. To mitigate the unacceptable computational cost, a variety of unlearning methods without retraining have also been proposed to approximate the forgetting effect of model retraining [Kong and Alfeld, 2022; Golatkar *et al.*, 2019; Graves *et al.*, 2020; Chundawat *et al.*, 2022b; Tarun *et al.*, 2021; Chundawat *et al.*, 2022a; Tarun *et al.*, 2022; Foster *et al.*, 2024; Baumhauer *et al.*, 2022], but they often require full access to training data and still incur high computational overheads [Kurmanji *et al.*, 2023].

To address the limitations stated above, recently proposed unlearning methods [Chen *et al.*, 2023; Cha *et al.*, 2024] fine-tune the target model with re-labeled forgetting data. The forgetting data are re-labeled to their neighboring classes, and this re-labeling technique can fulfill efficient unlearning promisingly by accessing the forgetting data only. However, existing re-labeling based methods still suffer from the following limitations. While not explicitly specified in their design, they primarily follow a single-step re-labeling scheme where all forgetting data are re-labeled to their neighboring classes only once based on the original model. This scheme fails to consider the individual relationship between a forgetting data sample and the model decision boundaries. As a result, this coarse-grained re-labeling scheme can assign a forgetting sample to a class that is far away from the one it should belong to if retraining is adopted for unlearning. Besides, these methods only re-label the forgetting data with hard labels. This forces the corresponding forgetting data to have a significant correlation only with the re-labeled class but low correlations with the entire class distribution in feature space. Then, only one of the classification regions will significantly affect the decision boundary around the forgetting data. Hence, these methods will result in abrupt reconstruction of the decision boundaries during fine-tuning.

In this paper, we propose an approach named *Self-Unlearning with Layered Iteration (SULI)* to address the

*Corresponding Author. Contact xuyun.zhang@mq.edu.au or hongsheng.hu@newcastle.edu.au

aforementioned limitations and improve unlearning efficacy further. The basic idea is to achieve an iterative fine-grained re-labeling process for the forgetting data, where the model is continuously fine-tuned and employed for label prediction in turn during unlearning. Specifically, the approach consists of two parts: the *layered iteration strategy* and the *selective probability adjustment (SPA)*. The layered iteration strategy progressively refines the decision boundaries from the high entropy region to the low entropy region. This adaptive approach ensures smooth boundary transitions while preserving model stability and utility. SPA re-labels the forgetting data with the soft-labels by calculating the probability distribution of the corresponding forgetting sample. SULI can substantially maintain model utility, significantly reduce the computational cost, and enhance data privacy preservation.

Contributions. We summarize our contributions as follows:

- We propose a novel unlearning framework named *Self-Unlearning with Layered Iteration (SULI)*. SULI modifies the single-step strategy with the *layered iteration strategy*, refining decision boundaries by prioritizing high-uncertainty samples in the early stages of iteration. The iteration framework achieves smooth and accurate boundary adjustments, improving unlearning effectiveness.
- SULI employs the *Selective Probability Adjustment (SPA)* strategy to tackle the problem of re-labeling forgetting data with hard labels. SPA calculates a new soft label for each sample to reduce the influence of data instances.
- We evaluate our approaches on three benchmark datasets against the state-of-the-art baselines, using four metrics to assess unlearning effectiveness, utility, privacy, and efficiency, with a detailed ablation study. Experimental results show that SULI achieves superior performance.

2 Related Work

Machine Unlearning. Machine unlearning is increasingly essential for ensuring privacy in machine learning models, particularly when data must be deleted due to regulatory requirements and individual privacy concerns [Nguyen *et al.*, 2022]. The existing machine unlearning methods can be divided into two categories:

• **Exact Unlearning**, introduced by [Cao and Yang, 2015; Bourtoule *et al.*, 2021], ensures the model behaves like the deleted data were never part of training. This is typically achieved by retraining the model from scratch without the unlearned data, producing a parameter state unaffected by the forgetting data. Although effective, this method is computationally expensive and impractical for large-scale models or frequent deletion requests [Kurmanji *et al.*, 2023].

• **Approximate Unlearning** aims to address these inefficiencies in exact unlearning by adjusting model parameters directly to emulate retraining effects [Izzo *et al.*, 2021; Goldblum *et al.*, 2020; Golatkar *et al.*, 2019]. [Ginart *et al.*, 2019] proposed a probabilistic framework based on differential privacy principles, requiring output distributions of unlearned models to resemble retrained models closely. However, [Thudi *et al.*, 2022] argued that achieving specific parameter configurations does not always guarantee effective unlearning, as it may fail to remove all traces of the forgetting data. To address this limitation, [Goel *et al.*, 2022] advocated focusing on functional equivalence, which ensures that the unlearned model behaves similarly to the retrained model.

Method	Forgetting data	W/O retaining data	Class	Instance
Fisher Unlearning [Golatkar *et al.*, 2019]	✗	✗	✗	✓
Batches Unlearning [Graves *et al.*, 2020]	✗	✗	✗	✓
Fast Yet Unlearning [Tarun *et al.*, 2021]	✗	✗	✓	✓
Zero-shot Unlearning [Chundawat *et al.*, 2022b]	✓	✗	✓	✓
SSD [Foster *et al.*, 2024]	✓	✗	✓	✓
SCRUB [Kurmanji *et al.*, 2023]	✓	✗	✓	✓
L-CODEC [Mehta *et al.*, 2022]	✓	✗	✗	✓
Contrastive Label [Kim and Woo, 2022]	✓	✗	✓	✓
Bad Teacher [Chundawat *et al.*, 2022a]	✓	✗	✓	✓
Recoverable Forgetting [Ye *et al.*, 2022]	✓	✗	✓	✗
Random Label [Graves *et al.*, 2020]	✓	✗	✓	✓
UGradSL [Di *et al.*, 2024]	✓	✗	✓	✓
NegGrad [Golatkar *et al.*, 2019]	✓	✓	✓	✓
Boundary Unlearning [Chen *et al.*, 2023]	✓	✓	✓	✗
Adversarial Unlearning [Cha *et al.*, 2024]	✓	✓	✗	✓
Ours	✓	✓	✓	✓

Table 1: Comparison of data requirements and unlearning scope. ✓: yes; ✗: no. 'Forgetting data': requires access to the forgetting data. 'W/O retaining data': operates without any retaining data. 'Class'/'Instance': supports class-wise or instance-wise unlearning.

Machine Unlearning in Deep Learning. Unlearning in deep neural networks presents significant challenges due to their high dimensionality and complex architectures [Nguyen *et al.*, 2022]. Existing methods primarily focus on approximate unlearning and vary in reliance on retaining or forgetting data. Table 1 compares applicability to class and instance unlearning scenarios and whether using the forgetting or retaining data. Most existing approaches require access to the full dataset, whereas ours operates solely on forgetting data, supporting both class-wise and instance-wise unlearning.

Most unlearning methods still need access to the original training dataset, using the retaining data to help remove the influence of forgetting data while preserving model utility. Two-stage approaches, such as those by [Kim and Woo, 2022] and [Wu *et al.*, 2022], first unlearn forgetting data and then recover utility using retaining data, achieving robust results but incurring significant computational and storage costs. Methods like *Amnesiac Unlearning* [Graves *et al.*, 2020] and *UGradSL* [Di *et al.*, 2024] employ random label assignments or gradient-based techniques to erase forgetting data, while teacher-student frameworks, including *SCRUB* [Kurmanji *et al.*, 2023] and *Bad Teacher* [Chundawat *et al.*, 2022a], aim to unlearn data without sacrificing performance. Noise-based approaches, such as error-maximizing noise matrices [Tarun *et al.*, 2021] and *Batch Unlearning* [Graves *et al.*, 2020], remove targeted gradient updates, and *Zero-shot Unlearning* [Chundawat *et al.*, 2022b] replaces forgetting data with noise before updating the model. However, reliance on full dataset access imposes a large computational overhead. Other methods only rely on retaining or forgetting data, aiming to reduce computational costs while maintaining model utility. *Fisher Unlearning* [Golatkar *et al.*, 2019] uses noise updates guided by the Fisher Information Matrix (FIM), which, despite reducing dataset requirements, incurs high computational costs and risks significantly degrading model utility. Similarly, *Negative Gradient* [Golatkar *et al.*, 2019] applies gradient ascent to erase forgetting data, leading to substantial performance loss.

Recent advancements, including *Boundary Unlearning*

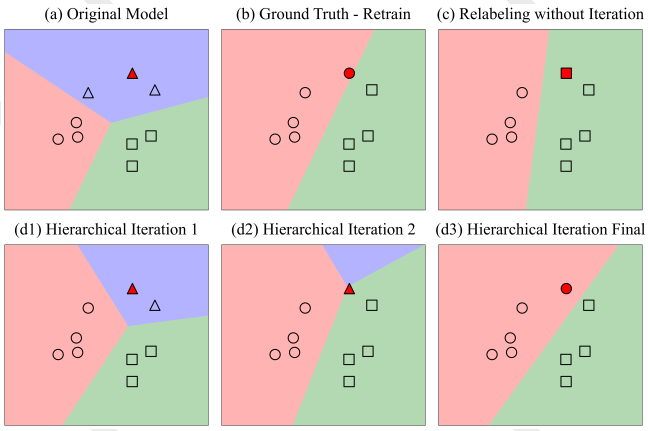


Figure 1: The boundary evolution of different unlearning methods is shown. (a) represents the original model with its decision boundaries. (b) represents the boundary of retraining from scratch. (c) shows re-labeling without iteration processes, highlighting less optimal boundary shifts. Figures (d1)-(d3) show the gradual boundary adjustments across hierarchical iterations. The red highlighted sample is eventually re-labeled to the same class, and the final result in (d3) closely matches the ground truth shown in (b).

[Chen *et al.*, 2023] and *ADV+IMP* [Cha *et al.*, 2024], aim to achieve unlearning by recalibrating decision boundaries based exclusively on forgetting data, thereby eliminating the reliance on retaining data. These methods implement a single-step re-labeling strategy, where forgetting data are reassigned to their nearest class boundary based on their position in the feature space. However, this static approach exhibits significant limitations, as it narrowly focuses on the nearest class and fails to account for the global structure of all classes within the model. Such a constrained perspective often results in the misclassification of forgetting data into irrelevant classes, leading to disruptions in the model's class representation and a reduction in overall performance. This issue becomes particularly pronounced in scenarios with complex decision boundaries or diverse class distributions, where class relationships are intricate and dynamic.

3 Method

3.1 Preliminaries

In our approach, the original training dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n}$ consists of samples $x_i \in \mathbb{R}^d$, representing d-dimensional feature vectors, and their corresponding labels $y_i \in \{1, 2, \ldots, K\}$, indicating their class among K unique categories. This dataset encompasses all categories foundational to our model's operation. We designate $\mathcal{D}_f$ as the subset of data intended for unlearning and $\mathcal{D}_r$ as the retaining set for maintaining accurate classification. These subsets are mutually exclusive and collectively exhaustive, satisfying $\mathcal{D}_r \cup \mathcal{D}_f = \mathcal{D}_{\text{train}}$ and $\mathcal{D}_r \cap \mathcal{D}_f = \emptyset$. Each sample-label pair from $\mathcal{D}_{\text{train}}$ is denoted as $(x, y) \in \mathcal{D}_{\text{train}}$, with $(x_f, y_f) \in \mathcal{D}_f$ and $(x_r, y_r) \in \mathcal{D}_r$ representing instances within the forgetting and retaining datasets, respectively.

The unlearning objective is to align the output distribution of the unlearned model f_{w_u} on the forgetting dataset $\mathcal{D}_f$ with that of the retrained model f_{w_r}, specifically aiming for:

$$P(f_{w_u}(x_f)) \approx P(f_{w_r}(x_f)).$$

Here, $f_w(x)$ denotes the model's output probabilities for input x under parameters w. The retrained model f_{w_r} is trained exclusively on the retaining dataset $\mathcal{D}_r$, thereby making predictions on $\mathcal{D}_f$ without prior exposure and relying on generalized knowledge from $\mathcal{D}_r$.

Algorithm 1 Self-Unlearning with Layered Iteration (SULI)

Input: Original model f_{w_0}, forgetting dataset $\mathcal{D}_f$, number of layers t, small positive value ϵ
Output: Unlearned model f_{w_t}
Compute entropy $H(x_f)$ for all $x_f \in \mathcal{D}_f$
Sort $\mathcal{D}_f$ in decreasing order of $H(x_f)$ and partition into t subsets $\{\mathcal{D}_{f1}, \mathcal{D}_{f2}, \ldots, \mathcal{D}_{ft}\}$
Set $w_0 \leftarrow$ parameters of original model
for each layer $i = 1$ to t **do**
  SPA Adjustment:
  for each $x \in \mathcal{D}_{fi}$ **do**
    Compute $P(y|x_f; w_{i-1})$
    Adjust $P'(y|x_f)$ using SPA with ϵ
  end for
  Model Update:
  Update w_t by minimizing:

$$w_i = \arg\min_{w} \sum_{x \in \mathcal{D}_{fi}} D_{\text{KL}}(P'(x_f) \parallel P(x_f; w_{i-1}))$$

end for
return f_{w_t}

3.2 Challenges of Single-Step Re-labeling

This paper focuses on unlearning in image classification tasks. Current re-labeling based unlearning methods face significant challenges in effectively controlling the impact of forgetting $\mathcal{D}_f$. Processing the entire forgetting dataset in a single step often leads to abrupt and inconsistent parameter updates, causing sharp shifts in decision boundaries and compromising model utility. The single-step approaches lack the granularity to refine decision boundaries adaptively, leading to coarse updates and misclassifications, particularly for samples close to the class center.

Figure 1 illustrates this issue. Subfigure (a) shows the model's decision boundaries before unlearning, while (b) depicts the ideal boundaries achieved through retraining on the retaining dataset $\mathcal{D}_r$. In contrast, the single-step approach shown in (c) abruptly adjusts boundaries by re-labeling all forgetting data, resulting in sharp shifts and deviations from the ideal boundaries. Samples near decision boundaries (highlighted in red) are prone to misclassification, weakening the model's ability to generalize and maintain stability.

3.3 Self-Unlearning with Layered Iteration (SULI)

The *Self-Unlearning with Layered Iteration (SULI)* framework addresses the limitations of single-step re-labeling by introducing a hierarchical iterative process, as illustrated in

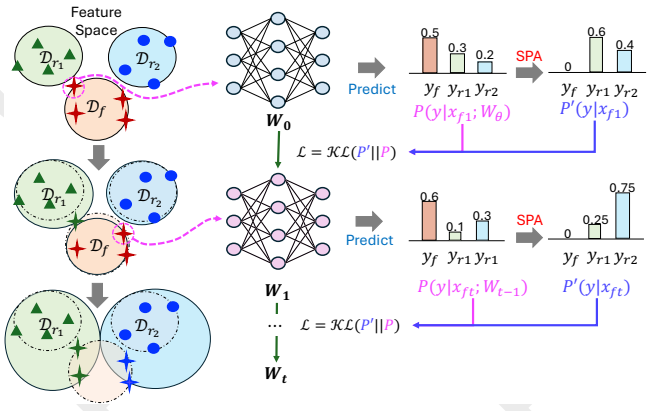


Figure 2: A hierarchical unlearning process in SULI proceeds as follows. Initially, samples in $\mathcal{D}_f$ exhibit strong affinities for their original classes (high confidence). Through iterative updates, SPA systematically reduces the model's dependence on $\mathcal{D}_f$ by redistributing probabilities to alternative classes. This gradual realignment prompts the model to treat $\mathcal{D}_f$ as if it had never been trained on it, relying solely on $\mathcal{D}_r$ for classification cues. As a result, decision boundaries shift smoothly, yielding predictions that mirror a fully retrained model while preserving performance on retaining data.

Figure 2. The unlearning process combines three key components: *Layered Iteration Framework*, *Entropy-Based Prioritization*, and *Selective Probability Adjustment (SPA)*. This synergy ensures that the decision boundaries of the model are progressively refined, aligning closely with the ideal retrained state while maintaining stability and generalization. Compared to single-step re-labeling, SULI introduces a structured and layered process to ensure that unlearning progress is in a stable and interpretable manner, avoiding significant disruptions to the overall decision structure.

Layered Iteration Framework. SULI partitions the forgetting dataset $\mathcal{D}_f$ into L independent subsets based on entropy, ensuring that high-uncertainty samples are addressed first. Each subset $\mathcal{D}_{ft}$ is processed iteratively, with the model parameters updated at each step t. The iterative update is guided by the following optimization objective:

$$w_{t+1} = \arg\min_{w} \mathcal{L}_t(w),$$

where $\mathcal{L}_t(w)$ represents the loss function at the t-th iteration. This general form highlights the layered structure of SULI while deferring the specific details of $\mathcal{L}_t(w)$ to subsequent sections. As shown in Figure 2, the hierarchical structure ensures that the model's parameters are updated iteratively in a structured manner. The retaining data $\mathcal{D}_r$ anchors the decision boundaries, while the forgetting data $\mathcal{D}_f$ undergoes selective adjustments. This layered approach mitigates abrupt changes and performs a stable transition to boundaries.

Entropy-based Prioritization. To determine the unlearning order of samples in $\mathcal{D}_f$, we compute entropy for each $\mathcal{D}_f$:

$$H(x_f) = -\sum_{i=1}^{K} P(y_i \mid x_f; w_0) \log\big(P(y_i \mid x_f; w_0)\big),$$

where $P(y_i \mid x_f; w_0)$ is the model's predicted probability for class y_i. Higher-entropy samples, typically near decision boundaries, are addressed first. Let $\{\mathcal{D}_{f1}, \mathcal{D}_{f2}, \ldots, \mathcal{D}_{ft}\}$ denote the resulting subsets in descending order of entropy. This prioritization directs the model to handle boundary-adjacent samples early, thereby minimizing their influence on subsequent iterations and enhancing the stability and efficiency of unlearning.

Selective Probability Adjustment (SPA). The SPA mechanism modifies the output probabilities of the model for $\mathcal{D}_f$ to simulate the effect of unlearning. For each sample $x_f \in \mathcal{D}_f$, the adjusted probability distribution $P'(y \mid x_f)$ is defined as:

$$P'(y|x) = \begin{cases} \epsilon, & \text{if } y = y_f, \\ \dfrac{P(y|x; w_0)}{S} \times (1-\epsilon), & \text{if } y \neq y_f, \end{cases}$$

where $S = \sum_{y' \in \mathcal{Y} \setminus \{y_f\}} P(y'|x; w_0)$ ensures normalization, and ϵ represents the unlearning degree. A detailed discussion on the choice and effect of ϵ is provided in Appendix A.

The iterative refinement process is formalized as:

$$w_t = \arg\min_{w} \sum_{x_{ft} \in \mathcal{D}_{ft}} D_{\mathrm{KL}}(P'(y \mid x_{ft}) \parallel P(y \mid x_{ft}; w_{t-1})).$$

This ensures that the model's parameters w_t are updated to minimize the divergence between the adjusted probabilities $P'(y \mid x_{ft})$ and the current model predictions $P(y \mid x_{ft}; w_{t-1})$. The process is repeated for each subset $\mathcal{D}_{ft}$, progressively refining the model's decision boundaries.

Self-Unlearning (SU). As a baseline, we introduce a straightforward, non-iterative approach, *Self-Unlearning (SU)*, which facilitates the unlearning process by treating the forgetting dataset $\mathcal{D}_f$ as unseen. This is achieved by modifying each sample's output distribution in a single pass using the SPA algorithm, effectively transforming $\mathcal{D}_f$ into data that the model perceives similarly to unseen data. In SU, given an instance $x_f \in \mathcal{D}_f$ and its adjusted probability distribution $P'(y \mid x_f)$, derived using the SPA algorithm, the SU approach updates the model through a single phrase of fine-tuning. The updated w' are obtained by minimizing the distillation loss:

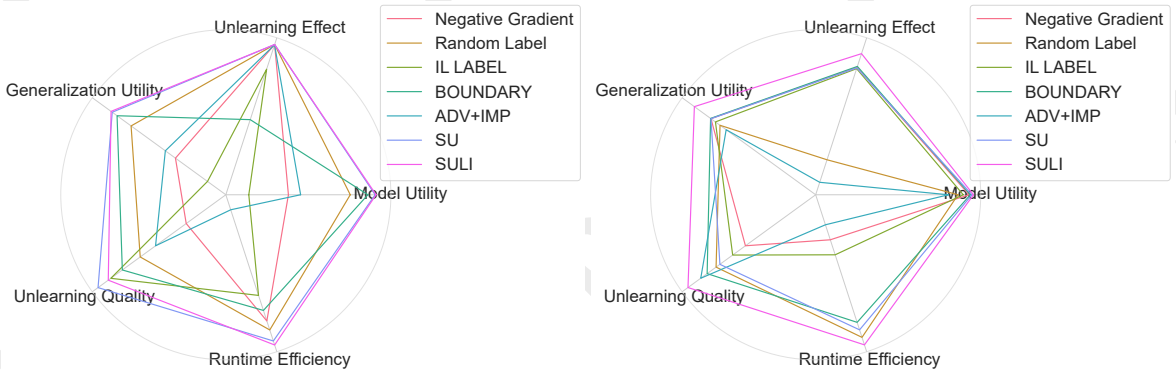
$$w' = \arg\min_{w} \sum_{x_f \in \mathcal{D}_f} D_{\mathrm{KL}}(P'(y \mid x_f) \parallel P(y \mid x_f; w_0)),$$

where D_{KL} is the KL-divergence, $P'(y \mid x_f)$ is the SPA-adjusted probability distribution, and $P(y \mid x_f; w_0)$ is the model's output distribution under w_0.

4 Experiments

Datasets. We follow the previous works [Chen *et al.*, 2023; Cha *et al.*, 2024] and use three datasets: CIFAR-10 [Krizhevsky, 2009], VGGFace2 [Cao *et al.*, 2018], and UTKFace [Zhang *et al.*, 2017]. These datasets are selected to evaluate different unlearning methods, including object recognition, face recognition, and age classification. Detailed dataset information is provided in Appendix B.

Models. Our experiments employ different models suitable for each dataset and train all models from scratch. For CIFAR-10 [Krizhevsky, 2009], we use a ResNet-18 model [He *et al.*, 2016]. For the UTKFace dataset [Zhang



(a) Class-wise Unlearning (b) Instance-wise Unlearning

Figure 3: Radar charts comparing the effectiveness of various unlearning methods across five objectives on CIFAR-10 datasets: maintaining model utility, effectiveness of data unlearning, ability to handle new data, thoroughness in removing data, and efficiency of the unlearning process. SULI maintains high model utility, ensuring the model's performance on retaining and unseen data remains robust after unlearning.

Dataset	Metric	Original	Retrain	NegGrad	Random Label	Initial Label	Boundary	ADV+IMP	SU (Ours)	SULI (Ours)
CIFAR-10	$A\mathcal{D}_r$ (↑)	100	100±0.0	29.62±2.65	78.77±2.47	14.54±1.34	89.94±0.17	47.35±2.94	94.54±0.29	**96.03±0.12**
	$A\mathcal{D}_f$ (↓)	100	0±0.0	**0.0±0.0**	**0.0±0.0**	9.69±0.03	12.54±0.0	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**
	$A\mathcal{D}_{tr}$(↑)	95	92.26±0.11	39.43±2.27	74.14±2.44	14.35±1.16	85.12±0.14	47.34±2.77	88.89±0.27	**89.81±0.17**
	$A\mathcal{D}_{tf}$(↓)	94.98	0.0±0.0	**0.0±0.0**	0.02±0.04	5.20±2.93	14.06±0.34	0.02±0.04	**0.0±0.0**	**0.0±0.0**
UTKFace	$A\mathcal{D}_r$ (↑)	99.97	99.96±0.04	86.79±3.11	84.33±3.31	72.45±6.12	94.08±0.98	78.52±2.52	96.86±0.24	**98.42±0.11**
	$A\mathcal{D}_f$ (↓)	99.95	0.0±0.0	**0.0±0.0**	**0.0±0.0**	21.84±3.09	8.13±0.12	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**
	$A\mathcal{D}_{tr}$(↑)	83.04	83.21±0.09	75.91±1.71	76.65±4.69	66.37±2.53	75.85±1.18	77.89±3.24	68.78±0.19	**79.81±0.13**
	$A\mathcal{D}_{tf}$(↓)	82.16	0.0±0.0	**0.0±0.0**	0.16±0.12	17.56±3.58	9.61±3.97	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**
VggFace2	$A\mathcal{D}_r$ (↑)	100	100±0.0	84.01±0.19	83.10±3.78	90.46±0.37	93.92±0.08	56.62±3.12	95.65±0.25	**97.51±0.21**
	$A\mathcal{D}_f$ (↓)	100	0±0.0	**0.0±0.0**	**0.0±0.0**	17.25±0.13	6.96±0.06	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**
	$A\mathcal{D}_{tr}$(↑)	80.95	81.12±0.02	60.85±0.21	63.60±3.72	64.62±0.54	71.96±0.14	41.11±2.56	72.49±0.16	**72.62±0.14**
	$A\mathcal{D}_{tf}$(↓)	81.48	0.0±0.0	**0.0±0.0**	**0.0±0.0**	9.52±0.34	3.61±0.19	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**

Table 2: Comparison of accuracy performance in class-wise unlearning.

et al., 2017], focusing on age prediction and demographic analysis, we adopt the All-CNN model [Springenberg *et al.*, 2014]. For the VGGFace2 dataset [Cao *et al.*, 2018], involving complex face recognition challenges, we utilize a ResNet-50 model [He *et al.*, 2016]. Detailed configurations and environment are provided in Appendix C.

Baselines. In this study, we select state-of-the-art machine unlearning methods as our baseline: (*a*) The Original model is trained on $\mathcal{D}_{train}$. (*b*) The Retrained model is retrained on $\mathcal{D}_r$ from scratch. (*c*) 'Negative Gradient' (NegGrad)[Golatkar *et al.*, 2019] modifies the original model by fine-tuning on $\mathcal{D}_f$ through gradient ascent . (*d*) 'Random Label' (RL) [Graves *et al.*, 2020] assigns arbitrary new labels to $\mathcal{D}_f$ and fine-tunes the network using these random labels. (*e*) 'Initial Label' (IL) [Chundawat *et al.*, 2022a] involves re-labeling $\mathcal{D}_f$ with new labels generated by an initial model and fine-tuning model with new labels. (*f*) 'Boundary Shrink' (Boundary) creates adversarial examples from $\mathcal{D}_f$ and assigns adversarial labels to induce boundary contraction towards disparate classes [Chen *et al.*, 2023]. (*g*) 'Adversarial Unlearning' (ADV+IMP) [Cha *et al.*, 2024] uses adversarial examples to re-label and weighted importance to update parameters.

Metrics. We adopt the following metrics to evaluate:

- *Accuracy on the unlearning and retaining datasets* ($A_{\mathcal{D}_f}$ and $A_{\mathcal{D}_r}$): This assesses the method's capability to unlearn without compromising the overall model performance.
- *Accuracy on the test datasets* ($A_{\mathcal{D}_t}$): This metric analyzes the model's generalization performance, further subdivided into unlearning test set ($A_{\mathcal{D}_{t_f}}$) and retaining test set ($A_{\mathcal{D}_{t_r}}$).
- *Membership Inference Attack (MIA)*: MIA assesses unlearning effectiveness by measuring how much information about the forgetting data remains in the model. We perform MIA techniques from [Kurmanji *et al.*, 2023]. Distance metrics like *Activation Distance* are used to measure unlearning success, but they fail to reflect actual outcomes due to inherent neural network variability [Hayes *et al.*, 2024].
- *Runtime*: This metric evaluates the method's temporal efficiency, providing a comparative computational cost analysis across various unlearning methods.

Implement Settings and Unlearning Tasks. Our experimental environment includes an NVIDIA RTX 4070 GPU, Python 3.11, and PyTorch 2.1.1. We utilize the ADAM optimizer [Kingma and Ba, 2014] with carefully selected learning rates optimized for both class-wise and instance-wise unlearning tasks. To ensure consistent evaluation, the same set of forgetting data ($\mathcal{D}_f$) is applied across all methods. We perform a grid search (the results are shown in appendix D) to optimize the hyperparameter t within the range [1, 25], selecting $t = 2$ for all experiments as it balances model util-

		CIFAR-10				UTKFace				VggFace2			
		n = 64	n = 128	n = 256	n = 512	n = 64	n = 128	n = 256	n = 512	n = 64	n = 128	n = 256	n = 512
$\mathcal{D}_r$ (↑)	Original	100	100	100	100	99.82	99.82	99.86	99.93	99.07	99.06	99.06	99.05
	Retrain	100	100	100	100	99.82	99.82	99.79	96.43	100	99.81	99.56	99.61
	Negative Gradient	99.86	99.18	96.39	95.39	92.23	86.47	75.66	67.71	95.23	83.66	84.22	82.48
	Random Label	97.21	95.10	90.33	77.10	89.49	84.47	76.64	67.37	94.02	87.62	79.97	78.57
	Initial Label	99.23	81.96	98.95	96.01	99.59	88.13	79.82	73.17	93.13	86.72	84.19	81.44
	Boundary Shrink	99.56	99.87	99.49	94.86	99.28	97.39	96.48	65.67	96.47	91.20	88.71	83.70
	Adv+Imp	92.05	81.96	82.82	78.76	96.91	87.48	82.86	77.60	88.43	83.73	81.39	78.84
	SU (Ours)	99.97	99.76	99.15	96.77	99.24	98.21	95.31	93.79	97.10	93.13	89.50	84.26
	SULI (Ours)	**99.99**	**99.91**	**99.68**	**97.27**	**99.29**	**98.32**	**97.57**	**96.13**	**98.12**	**96.93**	**93.92**	**91.49**
$\mathcal{D}_f$	Original	100.0	100	100	100	100.0	100.0	99.61	99.64	100	100	99.21	99.21
	Retrain	85.06	85.6	87.11	85.74	81.53	81.46	81.38	82.44	81.25	83.94	84.23	86.73
	Negative Gradient	82.81	85.16	85.55	85.55	81.29	80.47	80.13	81.36	83.74	83.82	83.79	83.71
	Random Label	82.19	81.88	83.75	80.35	80.13	80.72	80.19	80.71	83.70	83.71	83.72	83.68
	Initial Label	85.94	85.47	83.34	82.38	81.25	81.11	79.55	80.89	83.75	83.78	83.80	83.69
	Boundary Shrink	85.94	85.16	85.77	84.67	80.56	81.04	81.33	81.34	83.76	83.77	83.73	83.68
	Adv+Imp	85.94	85.47	82.62	79.01	81.81	80.46	81.03	80.08	83.75	83.72	83.76	83.68
	SU (Ours)	82.81	84.38	85.55	82.97	81.12	80.63	80.46	81.03	83.71	83.69	83.75	83.72
	SULI (Ours)	85.94	85.94	85.16	84.30	81.56	81.71	80.5	80.13	83.72	83.73	83.70	83.74
$\mathcal{D}_{test}$ (↑)	Original	86.34	86.34	86.34	86.34	82.96	82.96	82.96	82.96	84.52	84.52	84.52	84.52
	Retrain	85.2	86.15	85.67	86.09	81.41	82.51	82.86	82.31	83.26	83.52	84.56	85.27
	Negative Gradient	85.1	82.64	79.82	76.32	77.49	72.83	63.61	61.63	77.63	72.13	74.56	73.43
	Random Label	78.87	77.82	73.25	68.65	73.47	64.72	55.39	51.39	77.07	76.41	71.86	70.24
	Initial Label	84.42	69.54	81.15	76.96	81.73	75.54	62.79	58.12	75.48	74.18	69.27	64.35
	Boundary Shrink	83.98	83.89	83.07	78.78	81.47	79.43	78.89	70.45	76.33	72.29	69.29	64.03
	Adv+Imp	75.19	69.54	67.08	66.87	79.69	74.03	71.44	69.71	75.72	73.21	69.53	65.30
	SU (Ours)	85.10	83.76	81.84	79.30	82.53	80.72	79.04	74.36	79.27	74.41	76.29	71.29
	SULI (Ours)	**85.37**	**84.33**	**83.78**	**80.44**	**82.93**	**81.14**	**79.68**	**78.65**	**82.19**	**79.97**	**78.46**	**77.56**

Table 3: Evaluation results instance-wise unlearning of varying instance counts on CIFAR-10, UTKFace, and VggFace2.

Method	CIFAR-10		UTKFace		VGGFace2	
	Class	Instance	Class	Instance	Class	Instance
Original	0.57 ± 0.04	0.61 ± 0.02	0.63 ± 0.02	0.62 ± 0.03	0.63 ± 0.02	0.64 ± 0.02
Retrain	$\mathbf{0.51 \pm 0.01}$	$\mathbf{0.50 \pm 0.01}$	$\mathbf{0.51 \pm 0.01}$	$\mathbf{0.50 \pm 0.01}$	$\mathbf{0.50 \pm 0.02}$	$\mathbf{0.51 \pm 0.01}$
NegGrad	0.62 ± 0.05	$\mathbf{0.50 \pm 0.03}$	0.63 ± 0.04	0.56 ± 0.02	0.37 ± 0.02	0.43 ± 0.03
RLabel	0.54 ± 0.01	$\mathbf{0.52 \pm 0.03}$	0.55 ± 0.03	0.56 ± 0.04	0.57 ± 0.07	0.55 ± 0.04
ILabel	$\mathbf{0.49 \pm 0.01}$	$\mathbf{0.47 \pm 0.03}$	0.57 ± 0.07	0.46 ± 0.09	0.83 ± 0.05	0.63 ± 0.06
Boundary	0.56 ± 0.01	$\mathbf{0.53 \pm 0.03}$	0.62 ± 0.07	0.57 ± 0.04	0.61 ± 0.09	0.58 ± 0.04
ADV+IMP	0.57 ± 0.03	$\mathbf{0.49 \pm 0.04}$	0.45 ± 0.03	0.46 ± 0.02	0.44 ± 0.02	0.46 ± 0.04
SU (Ours)	$\mathbf{0.52 \pm 0.01}$	$\mathbf{0.48 \pm 0.03}$	$\mathbf{0.52 \pm 0.01}$	$\mathbf{0.52 \pm 0.02}$	$\mathbf{0.53 \pm 0.03}$	$\mathbf{0.52 \pm 0.02}$
SULI (Ours)	$\mathbf{0.51 \pm 0.01}$	$\mathbf{0.50 \pm 0.01}$	$\mathbf{0.51 \pm 0.01}$	$\mathbf{0.50 \pm 0.01}$	$\mathbf{0.49 \pm 0.02}$	$\mathbf{0.51 \pm 0.01}$

Table 4: Unlearning quality comparison of different methods. An MIA accuracy close to 0.5 indicates that the unlearning method has perfectly unlearned $\mathcal{D}_f$.

Metrics (s)	**ADV+IMP**	**Ilabel**	**Boundary**	**NegGrad**	**RLabel**	**SU (Ours)**	**SULI (Ours)**
Class Runtime	107.16	45.76	31.66	19.36	10.36	3.68	**0.93**
Instance Runtime	3.67	1.33	0.98	0.81	0.55	0.28	**0.21**

Table 5: SULI demonstrates the lowest runtime for both class and instance-level unlearning, showcasing its computational efficiency.

ity and unlearning effectiveness. Our experiments cover two primary unlearning scenarios: class-wise unlearning, where early stops when the model's accuracy on $\mathcal{D}_f$ approaches zero, and instance-wise unlearning, where unlearning ceases when the model's accuracy on $\mathcal{D}_f$ matches that on a 1% reference dataset. For both scenarios, we designate each class as $\mathcal{D}_f$ and report averages and standard deviations over five seeds to ensure reproducibility. Detailed implementation setups are provided in Appendix E.

Trade-offs. Achieving effective machine unlearning involves balancing multiple objectives: unlearning quality, model utility, unlearning effectiveness, and runtime efficiency. These objectives inherently involve trade-offs; for instance, maximizing unlearning quality might reduce efficiency or degrade performance on retaining data ($\mathcal{D}_r$) [Kurmanji *et al.*, 2023]. SULI is designed to navigate these trade-offs effectively, ensuring forgetting data ($\mathcal{D}_f$) is treated as unseen while maintaining strong performance on retaining data. It consistently outperforms other methods across multiple metrics (see Figure 3). Detailed discussions are provided in Appendix F.

Class-wise Accuracy. In class-wise unlearning, the goal is to reduce the model's accuracy on forgetting data ($A_{\mathcal{D}_f}$) and $A_{\mathcal{D}_{t_f}}$ to zero while preserving high accuracy on ($A_{\mathcal{D}_r}$). Table 2 shows that Model Retraining and NegGrad eliminate $A_{\mathcal{D}_f}$ but harm $A_{\mathcal{D}_r}$. Initial Label and Boundary Shrink methods struggle to sufficiently reduce $A_{\mathcal{D}_f}$ and also degrade $A_{\mathcal{D}_r}$. In contrast, ADV+IMP, SU and SULI successfully bring $A_{\mathcal{D}_f}$ and $A_{\mathcal{D}_{t_f}}$ to zero. Notably, SULI preserves the highest $A_{\mathcal{D}_r}$ and maintains the model utility.

Instance-wise Accuracy. In instance-wise unlearning, the goal is to align the model's accuracy on forgetting data ($\mathcal{D}_f$) with its performance on unseen reference data, ensuring fairness and generalizability. For simplicity, small standard deviations are omitted in subsequent instance-wise tables. Table 3 shows that while all methods reduce $\mathcal{D}_f$ accuracy, SU and SULI outperform others by achieving unlearning targets while maintaining high accuracy on $\mathcal{D}_r$ and $\mathcal{D}_{\text{test}}$.

Privacy Protection and Unlearning Quality Comparison. Effective unlearning is essential for robust privacy, preventing adversaries from using Membership Inference Attacks (MIA) to distinguish removed data from data never in the model. Following Kurmanji et al. [Kurmanji *et al.*, 2023], we train a binary classifier on loss values from $\mathcal{D}_f$ and $\mathcal{D}_{\text{test}}$. An ac-

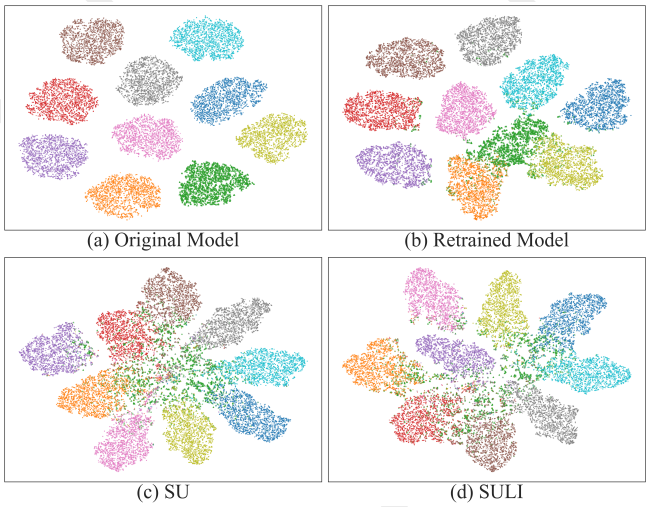


Figure 4: The visualization of feature space in different models on CIFAR10. The solid dots in various colors represent $\mathcal{D}_r$, and the green triangle represents the $\mathcal{D}_f$.

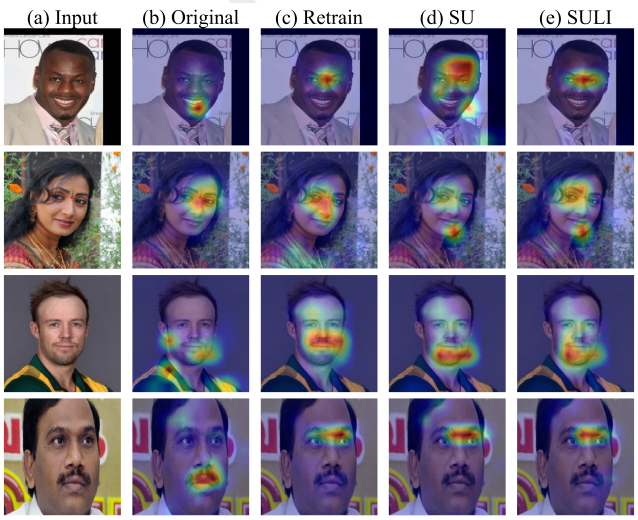


Figure 5: Gradient-weighted class activation mapping of the target models before and after the unlearning.

curacy nearing 50% indicates strong privacy protection, as the classifier fails to differentiate between these datasets. To ensure consistent data distributions in class-wise unlearning, $\mathcal{D}_f$ and $\mathcal{D}_{t_f}$ share the same categories. We then use ten-fold cross-validation to derive average MIA accuracies, minimizing outlier effects. Table 4 compares unlearning methods across class- and instance-level forgetting, revealing that SULI remains near the ideal 50% attack success rate, thus effectively obscuring data membership. Their hierarchical iteration and *Selective Probability Adjustment* (SPA) enable controlled, seamless unlearning with minimal utility loss. Further details appear in Appendix G.

Computational Complexity. Table 5 compares the runtime efficiency of various unlearning methods. SULI is the fastest method, requiring significantly less runtime than all other baselines (Table 5). In particular, it is 3 times faster than the adversarial re-labeling methods, which spend substantial time on iterative adversarial example generation. By contrast, SULI updates the model solely based on output distributions, avoiding the costly fine-grained perturbation steps. Overall, SULI provides the best balance of unlearning efficacy, privacy, and efficiency. Let $|\mathcal{D}_f|$ be the number of samples to be forgotten and K be the number of classes. SULI's total cost can be approximated as $\mathcal{O}(|\mathcal{D}_f| \times K)$, reflecting that it primarily involves computing probabilities and redistributing them for each sample-class pair, rather than performing computationally expensive adversarial example generation. Details are provided in Appendix H.

Visualization of Feature Space. To enable a more intuitive observation of how the decision boundary changes after the machine unlearning, we visualize the decision boundary of the target model's feature space using the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique [van der Maaten and Hinton, 2008]. The t-SNE maps high-dimensional data to a low-dimensional space through probability distributions, facilitating the visualization of complex data structures in graphs [van der Maaten and Hinton, 2008]. In figure 4 (d), we observe that the green triangles representing $\mathcal{D}_f$ are successfully dispersed across adjacent classes, effectively diluting their association with the original class. Concurrently, the clustering of other categories (depicted as dots) remains stable, and the decision boundaries between $\mathcal{D}_r$ classes are preserved without significant alteration. Dispersing $\mathcal{D}_f$ while keeping the decision boundaries of $\mathcal{D}_r$ demonstrates the effectiveness of SULI in unlearning target data without compromising the model's utility.

Class Activation Mapping. Gradient-weighted Class Activation Mapping (Grad-CAM) visualizes the decision-making processes of convolutional neural networks by highlighting image regions that most influence classification decisions [Selvaraju *et al.*, 2020]. Figure 5 compares attention maps across different models. For facial recognition, critical facial features are the eyes and mouth. SULI further disperses attention while concentrating on the face, albeit less on primary features and more on other facial regions. Despite the unlearning processes, the model continues to identify facial features by focusing on less discriminative areas. SULI's facial recognition attention pattern is similar to the retrain model.

5 Conclusion

We have introduced *Self-Unlearning with Layered Iteration (SULI)*, a machine unlearning framework that iteratively refines decision boundaries to remove specified data while maintaining accuracy in retaining data. By circumventing complete dataset access and avoiding abrupt re-labeling, SULI significantly reduces the influence of forgetting data while preserving model utility and privacy. The layered, stepwise design ensures model stability and effectively addresses the limitations of single-step boundary adjustments. In future work, we will explore scaling SULI to larger models and investigate theoretical guarantees for complete unlearning.

Acknowledgments

The work was partially supported by The State Key Laboratory of Novel Software Technology (KFKT2024A03).

References

[Baumhauer *et al.*, 2022] Thomas Baumhauer, Pascal Schöttle, and Matthias Zeppelzauer. Machine unlearning: Linear filtration for logit-based classifiers. *Machine Learning*, 111(9):3203–3226, 2022.

[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.

[Cao *et al.*, 2018] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[Carlini *et al.*, 2021] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[Cha *et al.*, 2024] Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers, 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. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7766–7775, 2023.

[Chundawat *et al.*, 2022a] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan S. Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. *ArXiv*, abs/2205.08096, 2022.

[Chundawat *et al.*, 2022b] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan S. Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2022.

[Di *et al.*, 2024] Zonglin Di, Zhaowei Zhu, Jinghan Jia, Jiancheng Liu, Zafar Takhirov, Bo Jiang, Yuanshun Yao, Sijia Liu, and Yang Liu. Label smoothing improves machine unlearning. *arXiv preprint arXiv:2406.07698*, 2024.

[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.

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

[Goel *et al.*, 2022] Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640*, 2022.

[Golatkar *et al.*, 2019] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9301–9309, 2019.

[Goldblum *et al.*, 2020] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Xiaodong Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:1563–1580, 2020.

[Graves *et al.*, 2020] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI Conference on Artificial Intelligence*, 2020.

[Hayes *et al.*, 2024] Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. *arXiv preprint arXiv:2403.01218*, 2024.

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

[Hu *et al.*, 2022] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022.

[Izzo *et al.*, 2021] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, pages 2008–2016. PMLR, 2021.

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

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[Kong and Alfeld, 2022] Zhifeng Kong and Scott Alfeld. Approximate data deletion in generative models. *ArXiv*, abs/2206.14439, 2022.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

[Kurmanji *et al.*, 2023] Meghdad Kurmanji, P. Triantafillou, and Eleni Triantafillou. Towards unbounded machine unlearning. *ArXiv*, abs/2302.09880, 2023.

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

[Nguyen *et al.*, 2022] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

[Selvaraju *et al.*, 2020] 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. *International journal of computer vision*, 128:336–359, 2020.

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

[Springenberg *et al.*, 2014] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[Tarun *et al.*, 2021] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan S. Kankanhalli. Fast yet effective machine unlearning. *IEEE transactions on neural networks and learning systems*, PP, 2021.

[Tarun *et al.*, 2022] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan S. Kankanhalli. Deep regression unlearning. *ArXiv*, abs/2210.08196, 2022.

[Thudi *et al.*, 2022] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4007–4022, 2022.

[van der Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[Voigt and Von dem Bussche, 2017] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[Wu *et al.*, 2022] Chen Wu, Sencun Zhu, and Prasenjit Mitra. Federated unlearning with knowledge distillation. *arXiv preprint arXiv:2201.09441*, 2022.

[Ye *et al.*, 2022] Jingwen Ye, Yifang Fu, Jie Song, Xingyi Yang, Songhua Liu, Xin Jin, Mingli Song, and Xinchao Wang. Learning with recoverable forgetting. In *European Conference on Computer Vision*, pages 87–103, 2022.

[Zhang *et al.*, 2017] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5810–5818, 2017.