Towards Natural Machine Unlearning

Zhengbao He **Tao Li** **Xinwen Cheng** **Zhehao Huang** **Xiaolin Huang**[*]
Department of Automation, Shanghai Jiao Tong University
{lstefanie, li.tao, xinwencheng, kinght_h, xiaolinhuang}@sjtu.edu.cn

Abstract

Machine unlearning (MU) aims to eliminate information that has been learned from specific training data, namely forgetting data, from a pre-trained model. Currently, the mainstream of existing MU methods involves modifying the forgetting data with incorrect labels and subsequently fine-tuning the model. While learning such incorrect information can indeed remove knowledge, the process is quite unnatural as the unlearning process undesirably reinforces the incorrect information and leads to over-forgetting. Towards more *natural* machine unlearning, we inject correct information from the remaining data to the forgetting samples when changing their labels. Through pairing these adjusted samples with their labels, the model will tend to use the injected correct information and naturally suppress the information meant to be forgotten. Albeit straightforward, such a first step towards natural machine unlearning can significantly outperform current state-of-the-art approaches. In particular, our method substantially reduces the over-forgetting and leads to stable performance in various unlearning settings, making it a promising candidate for practical machine unlearning.

1 Introduction

Modern machine learning models are essential in various applications [1, 2, 3, 4]. However, their heavy reliance on extensive data for training raises significant privacy concerns. The General Data Protection Regulations (GDPR) [5] emphasizes individuals' rights to request the deletion of their private data, leading to a surge of interests in machine unlearning (MU). MU aims to remove the influence of specific data in training set from a well-trained model. Recently, this field has garnered considerable attention not only for its contributions to privacy protection [6, 7, 8, 9] but also for its capacity to eliminate erroneous and sensitive data [10, 11, 12].

Current popular MU methods are primarily optimization-based, which achieve unlearning by fine-tuning the original model with manually crafted data. For instance, Amnesiac [13] optimizes the model using randomly labeled forgetting samples along with other remaining data. BadTeacher [14] relabels the forgetting samples with the predictions of a randomly initialized model as a "bad" teacher.

The name of "bad" teacher hits the essence of the above methods: they create *incorrect* information with the forgetting samples to fine-tune the model, compelling it to forget the correct information previously learned[1]. However, the incorrect information could be undesirably reinforced during the fine-tuning process. An obvious observation, illustrated in Fig. 1, is the so-called "over-forgetting", where after sufficient training, the accuracy on the forgetting samples (denoted as forgetting accuracy) is significantly lower than expected levels. This occurs because the incorrect information in these randomly relabeled instances appears quite unnatural in the view of remaining data [15]. The conflicts between the relabeled instances and remaining data cause the unlearned model to remember these forgetting samples even more firmly. To avoid over-forgetting, one can restrict the mobility scale by

[1] In this paper, we consider a training point as a "sample-label" pair, also referred to as an "instance". More details regarding the terminology can be found in Sec. 2.1.

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).

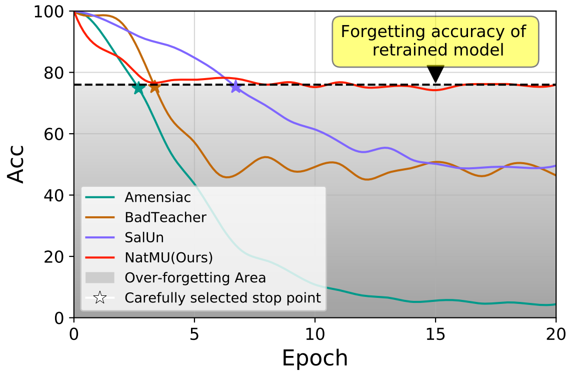


Figure 1: Accuracy comparison of different methods on forgetting samples. The experiments are taken on CIFAR-100 using ResNet18 under 1% random-subset unlearning setting. The dash line is the natural forgetting accuracy of retrained model, to which a smaller gap indicates better MU. The forgetting accuracy of other methods continuously decreases after crossing over the dash line, while ours can converge to that of the retrained model. Hence, to obtain a good MU performance, other methods may need to carefully stop training at a middle point, donated as ★. Best viewed in color.

identifying a parameter mask [16] or by carefully selecting a stopping point. However, this requires meticulous hyperparameter tuning, which is impractical in real-world applications.

To address the problem of over-forgetting, we contend that the unlearning process should be *natural*, minimizing the conflicts within the fine-tuning data. The most natural MU process is to retrain the model from scratch with the training data excluding the forgetting ones, which is the golden standard for MU [17, 18, 19]. This method ensures that all information used is correct, eliminating concerns about over-forgetting since the forgetting samples are excluded from the training process. However, retraining requires significant computational resources, making it impractical in many cases. The challenge, therefore, lies in balancing the inclusion of forgetting samples to enhance unlearning efficiency and the exclusion of incorrect information to maintain the process's naturalness.

In this paper, we introduce a novel method called "NatMU" that aims to facilitate a more natural unlearning by injecting correct information into the forgetting samples. Specifically, such correct information is extracted from the remaining data and then injected into forgetting samples for creating hybrid samples. The hybrid samples are subsequently assigned categories consistent with the injected information. Each hybrid sample consists of two distinct types of information: one from the forgetting sample, and the other from the remaining sample. By learning to pair the hybrid sample with the reassigned label, the connection between injected information and corresponding label is reinforced. This naturally suppresses model's response to the former type of information which is to be forgotten, thereby achieving effective unlearning. Since such reinforced connection inherently exists within the remaining set, NatMU achieves a more natural machine unlearning process.

NatMU takes the first step towards natural machine unlearning with the injection of correct information, which significantly reduces the conflicts within the fine-tuning data. The resulting model can maintain a natural generalization on forgetting samples, consistent with the predictions of retrained model. As a result, NatMU can narrow the forgetting accuracy's gap with the retrained model from 39.44% to 1.72% when unlearning sub-class "Vehicle2" on CIFAR-20. The natural property eliminates the need to stop training at unstable points for good performance. Consequently, NatMU demonstrates stable performance under various settings without changing hyperparameters, enabling the application of shared hyperparameters in real-world applications. For instance, when altering the forgetting ratio from 10% to 1%, NatMU maintains an average forgetting accuracy gap of only 2.03% across three datasets, while other methods exhibit a gap over 20%.

Our contributions can be summarized as follows:

- We first point out the unnatural property of the previous optimization-based MU methods, which leads to issues such as unnatural generalization on forgetting samples and impracticality.

- We propose an effective and efficient MU approach towards natural machine unlearning, named NatMU, which successfully addresses previous issues by injecting correct information into forgetting samples to remove the information to be forgotten.

- Extensive experiments on various datasets and multiple machine unlearning scenarios demonstrate that NatMU significantly narrows the performance gap with retrained model compared to previous approaches with strong robustness.

2 Towards Natural Machine Unlearning

2.1 Preliminaries

Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ be the training set containing inputs sample $\boldsymbol{x}_i \in \mathbb{R}^d$ with corresponding label $y_i \in \{1, 2, ..., K\}$. We denote a sample-label pair as an "instance" in this paper. Let forgetting set $\mathcal{D}^f \subset \mathcal{D}$ be the set of forgetting data and remaining set $\mathcal{D}^r = \mathcal{D} \setminus \mathcal{D}^f$ be the set of remaining data. The machine learning model can be represented as a parameterized function $f_\theta(\cdot) : \mathbb{R}^d \to \mathbb{R}^K$ with parameter θ. Let θ_o be the parameter of original model. The objective of approximate MU is to **narrow the performance gap between the unlearned and retrained model** in output space.

Mainstream machine unlearning work primarily focuses on two unlearning scenarios: (1) *full-class unlearning*, where $\mathcal{D}^f = \{(\boldsymbol{x}_i, y_i) \in \mathcal{D}, y_i = k\}$ contains all samples with label $y_i = k$. (2) *sub-class unlearning*, where $\mathcal{D}^f \subset \{(\boldsymbol{x}_i, y_i) \in \mathcal{D}, y_i = k\}$ contains a related subset of the samples labeled as k. (3) *random-subset unlearning*, where $\mathcal{D}^f$ is a random subset of $\mathcal{D}$. The size of $\mathcal{D}^f$ is determined by the forgetting ratio. Among these unlearning scenarios, random-subset unlearning presents the greatest challenge due to the intricate intertwining between $\mathcal{D}^f$ and $\mathcal{D}^r$.

2.2 Revisiting Previous MU Methods

Optimization-based MU algorithms demonstrate exceptional performance across various scenarios, especially in random-subset unlearning. These methods require fine-tuning the original model on a carefully crafted fine-tuning dataset, typically comprising two parts: one built from the remaining data to maintain generalization ability and the other from the forgetting data to facilitate unlearning. We name the instances in the latter part as **unlearning instances**. In Amnesiac [13], the fine-tuning dataset $D_{\text{AMN}} = \mathcal{D}^r \cup \mathcal{D}^f_{\text{RL}}$, where $\mathcal{D}^f_{\text{RL}} = \{(\boldsymbol{x}_i, y_i^{\text{rand}}), (\boldsymbol{x}_i, y_i) \in \mathcal{D}^f, y_i^{\text{rand}} \neq y_i\}$. BadTeacher [14] utilizes the original original f_{θ_o} and a randomly initialized model $f_{\theta_{\text{bad}}}$ to construct softly labeled fine-tuning dataset from $\mathcal{D}^r$ and $\mathcal{D}^f$ respectively. Its remaining part is constructed with the original model: $\mathcal{D}^r_{\text{ori}} = \{(\boldsymbol{x}_i, f_{\theta_o}(\boldsymbol{x}_i)), (\boldsymbol{x}_i, y_i) \in \mathcal{D}^r\}$. While the forgetting part uses a randomly initialized model: $\mathcal{D}^f_{\text{bad}} = \{(\boldsymbol{x}_i, f_{\theta_{\text{bad}}}(\boldsymbol{x}_i)), (\boldsymbol{x}_i, y_i) \in \mathcal{D}^f\}$. SalUn [16] improves Amnesiac by freezing certain model parameters according to a gradient-based weight saliency map.

To approximate the retrained model in the output space, the ideal approach would be to relabel the forgetting samples with their predictions on the retrained model as the ground truth labels. However, in real-world unlearning scenarios, these ground truth labels cannot be obtained. Therefore, to ensure the unlearning process can effectively remove the learned knowledge, previous works can only assign incorrect labels to these samples as a secondary priority. This line of works are based on a consensus that the information in an instance $(\boldsymbol{x}_i, y_i)$ is determined by both the sample and its corresponding label. When y_i changes, the instance's information also significantly changes [20]. Thus, learning such modified data that contain incorrect information can remove the learned knowledge from them.

However, as illustrated in Fig. 1, these methods demonstrate an "over-forgetting" issue. After sufficient training, the accuracy of classifying the forgetting samples as their original labels is significantly lower than that of retraining. The underlying challenge is that incorrect information, i.e., the forgetting samples with their incorrect labels, is undesirably reinforced during fine-tuning. The unlearning process is quite unnatural since it compels the unlearned model to learn incorrect information, which actually conflicts with the remaining data. This adversely affects the model's natural ability to generalize these samples, which the retrained model preserves. For example, in random-subset unlearning, most of the forgetting samples can still be correctly classified by the retrained model, even if they are not involved in the training. Essentially, this unlearning process alters the learned knowledge to incorrect knowledge, instead of removing it. In addition to forgetting accuracy, various metrics can be also employed to measure the differences between the unlearned and retrained models, which will be discussed in Sec. 3.1. Despite meticulous hyperparameter tuning can make one metric close to the retrained model, it cannot ensure that other metrics will also be close. Gradient-ascent based methods also face a similar challenge in determining the optimal point to stop optimization. Stopping too early may result in insufficient forgetting, while stopping too late can lead to the over-forgetting of the forgetting data.

Therefore, it is crucial to realize a more natural machine unlearning while ensuring its efficiency. On one hand, the forgetting samples should be included to enhance unlearning efficiency. On the other hand, excluding incorrect information ensures a natural machine unlearning process. In this paper, we propose a novel method, named NatMU, towards natural machine unlearning. The core idea of NatMU is to inject information extracted from the remaining data into the forgetting samples to reduce the conflicts in each modified instance.

2.3 Proposed Method

Overview. Our NatMU method can be decomposed into three steps. Firstly, we randomly select n distinctive instances from the remaining set for each forgetting sample $\boldsymbol{x}^f$. Secondly, a injecting function $\mathcal{T}$ is employed to inject the information of remaining sample $\boldsymbol{x}^r$ into $\boldsymbol{x}^f$ by blending them at the pixel level, generating an unlearning instance $(\mathcal{T}(\boldsymbol{x}^f, \boldsymbol{x}^r), y^r)$. Thirdly, by merging remaining set with all generated unlearning data, we fine-tune original model on the resulting fine-tuning dataset.

Selection of remaining instances. For a forgetting instance $(\boldsymbol{x}^f, y^f) \in \mathcal{D}^f$, the n selected remaining instances $\{(\boldsymbol{x}^r_j, y^r_j)\}_{j=1}^n$ should have distinctive categories from y^f to ensure the effectiveness of unlearning. These remaining instances inherently reflect the correct information of the remaining data's distribution. To prevent that the model from having a preference for mapping the forgetting sample to any reassigned category, these selected instances should also have different categories from each other. Moreover, to reduce the conflicts in unlearning instances $(\mathcal{T}(\boldsymbol{x}^f, \boldsymbol{x}^r_j), y^r_j)$, the categories of remaining instances should be relevant to the forgetting sample. To meet these requirements, we calculate the top n predicted categories for the forgetting sample on the original model, excluding the original category y^f. Then, we sequentially select one remaining instance for each category.

The injecting function $\mathcal{T}$. Motivated by data augmentation like MixUp [21] and CutMix [22], we opt for a straightforward approach where two samples are added pixel by pixel: $\mathcal{T}(\boldsymbol{x}^f, \boldsymbol{x}^r) = \boldsymbol{x}^f + \boldsymbol{x}^r$. To ensure that the blended sample numerically matches the original distribution, a weighting mask vector $\boldsymbol{m} \in [0,1]^d$ is introduced. This vector controls the contribution of two samples at each pixel as defined by $\mathcal{T}_{\boldsymbol{m}}(\boldsymbol{x}^f, \boldsymbol{x}^r) = \boldsymbol{x}^f \circ \boldsymbol{m} + \boldsymbol{x}^r \circ (\mathbf{1}_d - \boldsymbol{m})$, where $\circ$ represents element-wise multiplication, and $\mathbf{1}_d$ represents a d-dimensional vectors of ones. The weighting operation offers an additional advantage that $\boldsymbol{x}^f \circ \boldsymbol{m}$ can be regarded as a segment of the complete forgetting sample $\boldsymbol{x}^f$. Consequently, it makes the unlearning process more friendly since each hybrid sample forgets only a part of $\boldsymbol{x}^f$. We have developed four "gradual MixUp" weighting vectors $\{\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3, \boldsymbol{m}_4\}$, with symmetrical elements that vary gradually in different directions. To adjust the proportion of forgetting samples in the hybrid samples, a scaling factor δ is introduced, which scales the values of weighting vectors as follows:

$$\boldsymbol{m}_j^{\text{scaled}} = \text{clip}_{[0,1]}(\boldsymbol{\delta}_d + \boldsymbol{m}_j), \tag{1}$$

where $\boldsymbol{\delta}_d$ denotes a d-dimensional vector with all elements equal to δ and $\text{clip}_{[0,1]}(\cdot)$ denotes a function which truncates vector elements to $[0,1]$.

Constructing the fine-tuning dataset. For a forgetting instance $(\boldsymbol{x}^f_i, y^f_i) \in \mathcal{D}^f$, we select four remaining instances $\{(\boldsymbol{x}^r_j, y^r_j) \in \mathcal{D}^r\}_{j=1}^{n=4}$ according to the above rules. Then, we generate unlearning instances using the following formula:

$$\mathcal{D}^f_i = \{(\mathcal{T}_{\boldsymbol{m}_j^{\text{scaled}}}(\boldsymbol{x}^f_i, \boldsymbol{x}^r_j), y^r_j)\}_{j=1}^n. \tag{2}$$

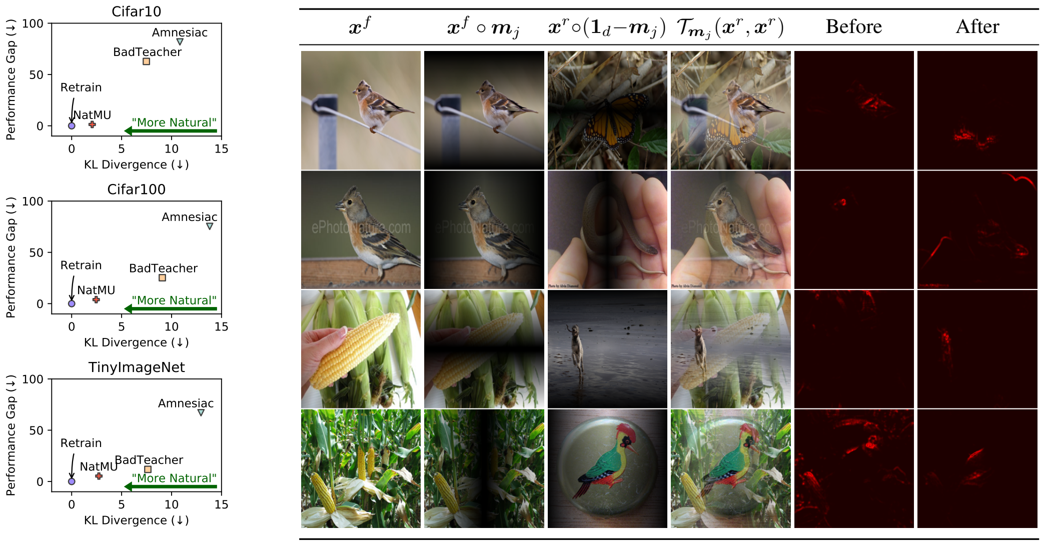
After executing the aforementioned operation on all forgetting samples, following [13, 16], we merge all the sets from these samples with the remaining set to compile the final fine-tuning dataset:

$$\mathcal{D}_{\text{Nat}} = \mathcal{D}^r \cup \mathcal{D}^f_{\text{forget}}, \text{ where } \mathcal{D}^f_{\text{forget}} = \mathcal{D}^f_1 \cup \mathcal{D}^f_2 \cup \cdots \mathcal{D}^f_{|\mathcal{D}^f|}. \tag{3}$$

Finally, we fine-tune the original model on the resulting dataset for several epochs, akin to other optimization-based methods.

2.4 Discussion

Unlearning mechanism. Given an unlearning instance $(\boldsymbol{x}^f \circ \boldsymbol{m} + \boldsymbol{x}^r \circ (\mathbf{1}_d - \boldsymbol{m}), y^r) \in \mathcal{D}_{\text{Nat}}$, it consists of two distinct parts of information: the information to be forgotten, denoted as $\boldsymbol{x}^f \circ \boldsymbol{m}$, and the injected information, denoted as $\boldsymbol{x}^r \circ (\mathbf{1}_d - \boldsymbol{m})$. In the original pre-trained model, the former information is connected to the forgetting label y^f, while the latter information is connected to the reassigned label y^r. During the fine-tuning of unlearning process, the unlearned model learns to pair



(a)

(b)

Figure 2: **(a)** Relationship between the performance gap on forgetting accuracy and KL divergence on different datasets. A smaller KL divergence indicates more natural MU. NatMU's KL divergence is much smaller than other methods, i.e., more natural, thus resulting in a smaller performance gap. **(b)** Visualization of our unlearning instances and their attention maps before and after unlearning calcualted with LRP [23]. After unlearning, the attention is shifted to the remaining information.

the unlearning sample with label y^r, reinforcing the latter connection. It naturally suppresses model's response to the forgetting information, since such response is harmful to make accurate predictions for unlearning instances. As a result, it effectively removes the learned knowledge in $\boldsymbol{x}^f \circ \boldsymbol{m}$.

Compared to previous label-modifying work, the reinforced information in NatMU is the connection between $\boldsymbol{x}^r \circ (\mathbf{1}_d - \boldsymbol{m})$ and y^r, which inherently exists in the remaining set. This prevents the model from establishing undesirable new associations between $\boldsymbol{x}^f \circ \boldsymbol{m}$ and y^r, thereby promoting a more natural unlearning process.

Fig. 2b visualizes the unlearning instances of NatMU and their attention maps before and after unlearning. We can see that different weighting masks capture different parts of forgetting samples in the second column. By comparing the attention maps before and after unlearning, model's attention is effectively shifted to the prominent positions of the remaining samples through unlearning. For example, in the first row, before unlearning the model focuses on the bird's wings in $\boldsymbol{x}^f \circ \boldsymbol{m}$. After unlearning, the model shifts it attention to the butterfly below the bird, which comes from $\boldsymbol{x}^r \circ (\mathbf{1}_d - \boldsymbol{m})$.

Natural property. The injected information is extracted from remaining data, therefore, the unlearning instances of NatMU align with the distribution of remaining data more closely. It significantly reduces the conflict between the unlearning instances and the remaining data. To quantify the degree of data conflict, we adopt the KL divergence [24] between the reassigned unlearning labels and retrained model's predictions on the unlearning samples. A smaller KL divergence indicates more natural unlearning, as the unlearned data aligns better with the retrained model's knowledge distribution. Since no unlearning samples are involved in retraining, we define its KL divergence as zero. In Fig. 2a, we demonstrate the average KL divergence of different methods over their unlearning instances. For Amnesiac, the unlearning instances are from $\mathcal{D}^f_{\mathrm{RL}}$. For BadTeacher, from $\mathcal{D}^f_{\mathrm{bad}}$; and for NatMU, from $\mathcal{D}^f_{\mathrm{forget}}$. We can see that NatMU has a much smaller KL divergence compared to other methods, indicating that its unlearning is more natural. Therefore, NatMU's performance is closer to that of the retrained model.

Preventing undesirable reinforcement. In previous work, incorrect information is reinforced during unlearning, leading to the over-forgetting problem. Specifically, the model establishes incorrect associations between the forgetting samples and their reassigned labels.

Although NatMU also modifies the forgetting labels, the existence of correct information from x^r can effectively prevent these incorrect associations. To verify this, we demonstrate the accuracy curves onf sample-label pairs $\{(\boldsymbol{x}^f \circ \boldsymbol{m}, y^r)\}$ and forgetting accuracy curves of different MU models trained with or without the correct information $\boldsymbol{x}^r \circ (\mathbf{1}_d - \boldsymbol{m})$ in Fig. 3. It can be seen that, without the injected correct information, the model quickly recognizes the partial forgetting sample $\boldsymbol{x}^f \circ \boldsymbol{m}$ as the reassigned class y^r. As a result, the forgetting accuracy faces a significant decrease. In contrast, the model trained with correct information consistently remains a low accuracy on $\{(\boldsymbol{x}^f \circ \boldsymbol{m}, y^r)\}$, showing that injecting correct information can effectively prevent undesirable reinforcement.

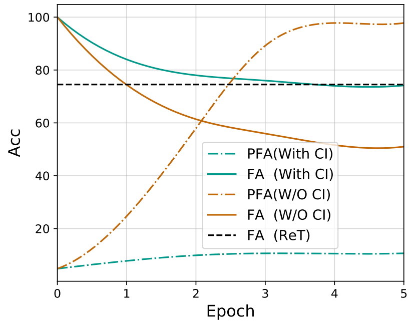


Figure 3: Accuracy curves on partial unlearning instances $\{(\boldsymbol{x}^f \circ \boldsymbol{m}, y^r)\}$ and forgetting accuracy of different MU models trained with/without correct information. Conducted on CIFAR-100 with a forgetting ratio of 1% using ResNet18. PFA: accuracy of classifying partial forgetting samples $\boldsymbol{x}^f \circ \boldsymbol{m}$ as random label y^r. FA: forgetting accuracy. CI: correct information. ReT: the retrained model.

The injected correct information form remaining data helps NatMU achieve a more natural machine unlearning, not forced unlearning as other label-modifying methods do. The natural generalization on forgetting samples is well preserved in NatMU. As a result, NatMU can achieve a smaller performance gap with the retrained model on multiple metrics. Moreover, the natural property gives NatMU greater robustness, allowing us to use nearly identical hyperparameters across different unlearning settings, making NatMU a promising candidate for practical MU.

3 Experiments

In this section, we evaluate NatMU against other baselines across different datasets, models, and unlearning settings. Then we demonstrate the performance of different methods when transferring the hyperparameters of one known setting to another unknown setting, which provides a hyperparameter determination way in real-world applications.

3.1 Setup

Datasets. NatMU is evaluated against other machine unlearning methods in the context of supervised image classification tasks using the CIFAR-10, CIFAR-20, CIFAR-100 [25], and TinyImageNet-200 [26] datasets. It is noteworthy that CIFAR-20 and CIFAR-100 are closely related. CIFAR-100 dataset consists of 20 superclasses, each containing 5 subclasses, resulting in a total of 100 classes. When considering only the superclasses, CIFAR-100 reduces to CIFAR-20.

Unlearning scenarios. Following [14] and [27], we evaluate MU methods across three unlearning scenarios: (1) *full-class unlearning*, (2) *sub-class unlearning*, and (3) *random-subset unlearning*. Models of various architectures are trained in different unlearning scenarios, including VGG16-BN [28], ResNet18 and ResNet34 [29].

Baselines. We compare NatMU with basic MU methods and multiple popular MU methods, including *Finetune* [30] which fine-tunes the original model on D^r, *gradient ascent (GA)* [18] which performs gradient ascent over D^f, *Amnesiac* [13], *BadTeacher* [14], *SalUn* [16], and *SSD* [27]. Some of these methods are designed for specific unlearning scenarios. Under different unlearning scenarios, we report only the methods that performed well.

Evaluation metrics. Following [14, 16, 27, 31, 32], we evaluate MU methods across three metrics: retain accuracy (**RA**), forgetting accuracy (**FA**) and **MIA** ratio [33]. Retain accuracy calculates the accuracy on test data, which measures the generalization ability. Forgetting accuracy evaluates the model's accuracy on the forgetting samples, which measures the effectiveness of MU. In full/sub-class unlearning, we calculate accuracy on samples of the forgetting class in training set and test set respectively, denoted as **FATrain** and **FATest**. MIA provides another view to evaluate the effectiveness of MU algorithms from the entropy of model outputs beyond forgetting accuracy. which is widely adopted in recent MU research. We also report the average performance gap with the retrained model

Table 1: Full-class unlearning on CIFAR-100 using ResNet18 with different forgetting classes. The results are given by $\boldsymbol{a}_{\pm \boldsymbol{b}}(\boldsymbol{c})$, where $\boldsymbol{a}$ donates the mean value, $\boldsymbol{b}$ donates the standard deviation, and $\boldsymbol{c}$ donates the performance gap with the retrained model over 5 independent trails. A smaller $\boldsymbol{c}$ means a better performance of MU methods.

Class	Metric	Retrain	Finetune [30]	GA [18]	Amnesiac [13]	BadTeacher [14]	SalUn [16]	SSD [27]	Ours
Rocket	RA	76.44$_{\pm0.33}$	76.36$_{\pm0.07}$(0.07)	71.93$_{\pm0.00}$(4.51)	76.87$_{\pm0.06}$(0.44)	76.88$_{\pm0.09}$(0.45)	76.97$_{\pm0.17}$(0.53)	76.53$_{\pm0.03}$(0.10)	76.63$_{\pm0.23}$(0.19)
	FATrain	0.00$_{\pm0.00}$	0.00$_{\pm0.00}$(0.00)	2.40$_{\pm0.00}$(2.40)	0.00$_{\pm0.00}$(0.00)	5.60$_{\pm0.85}$(5.60)	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)
	FATest	0.00$_{\pm0.00}$	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)	0.20$_{\pm0.40}$(0.20)	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)
	MIA	14.64$_{\pm0.77}$	6.08$_{\pm0.48}$(8.56)	6.60$_{\pm0.00}$(8.04)	8.68$_{\pm0.99}$(5.96)	0.00$_{\pm0.00}$(14.64)	7.04$_{\pm0.66}$(7.60)	0.88$_{\pm0.10}$(13.76)	11.52$_{\pm1.27}$(3.12)
	AvgGap	-	2.16	3.74	1.60	5.22	2.03	3.46	**0.83**
Sea	RA	76.44$_{\pm0.19}$	76.54$_{\pm0.28}$(0.09)	73.89$_{\pm0.00}$(2.56)	76.18$_{\pm0.28}$(0.26)	77.85$_{\pm0.10}$(1.40)	76.89$_{\pm0.17}$(0.45)	74.69$_{\pm0.22}$(1.75)	76.83$_{\pm0.06}$(0.39)
	FATrain	0.00$_{\pm0.00}$	0.00$_{\pm0.00}$(0.00)	1.40$_{\pm0.00}$(1.40)	0.00$_{\pm0.00}$(0.00)	15.76$_{\pm3.62}$(15.76)	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)
	FATest	0.00$_{\pm0.00}$	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)	25.60$_{\pm3.56}$(25.60)	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)	0.00$_{\pm0.00}$(0.00)
	MIA	26.84$_{\pm1.56}$	9.64$_{\pm0.87}$(17.20)	48.60$_{\pm0.00}$(21.76)	5.00$_{\pm1.02}$(21.84)	0.00$_{\pm0.00}$(26.84)	4.60$_{\pm0.87}$(22.24)	0.44$_{\pm0.20}$(26.40)	30.56$_{\pm0.54}$(3.72)
	AvgGap	-	4.32	6.43	5.53	17.40	5.67	7.04	**1.03**

Table 2: Sub-class unlearning on CIFAR-20 using ResNet18 with different forgetting sub-classes. The results are given by $\boldsymbol{a}_{\pm \boldsymbol{b}}(\boldsymbol{c})$, sharing the same format with Tab. 1.

Class	Metric	Retrain	Finetune [30]	GA [18]	Amnesiac [13]	BadTeacher [14]	SalUn [16]	SSD [27]	NatMU
Rocket	RA	84.94$_{\pm0.14}$	84.12$_{\pm0.27}$(0.81)	80.60$_{\pm0.00}$(4.34)	84.80$_{\pm0.09}$(0.14)	85.25$_{\pm0.05}$(0.32)	84.93$_{\pm0.07}$(0.00)	84.69$_{\pm0.12}$(0.24)	84.94$_{\pm0.25}$(0.00)
	FATrain	3.04$_{\pm0.34}$	23.88$_{\pm5.01}$(20.84)	1.80$_{\pm0.00}$(1.24)	2.60$_{\pm0.33}$(0.44)	15.24$_{\pm1.83}$(12.20)	3.96$_{\pm0.61}$(0.92)	0.64$_{\pm0.78}$(2.40)	6.44$_{\pm0.92}$(3.40)
	FATest	1.60$_{\pm0.49}$	20.40$_{\pm4.08}$(18.80)	1.00$_{\pm0.00}$(0.60)	2.40$_{\pm0.49}$(0.80)	5.00$_{\pm0.89}$(3.40)	3.20$_{\pm0.40}$(1.60)	1.20$_{\pm0.40}$(0.40)	3.60$_{\pm0.49}$(2.00)
	MIA	21.68$_{\pm2.08}$	4.20$_{\pm1.37}$(17.48)	15.40$_{\pm0.00}$(6.28)	0.00$_{\pm0.00}$(21.68)	0.00$_{\pm0.00}$(21.68)	0.00$_{\pm0.00}$(21.68)	8.68$_{\pm0.60}$(13.00)	16.84$_{\pm0.72}$(4.84)
	AvgGap	-	14.48	3.11	5.76	9.40	6.05	4.01	**2.56**
Sea	RA	84.66$_{\pm0.10}$	84.17$_{\pm0.16}$(0.49)	83.70$_{\pm0.00}$(0.96)	84.42$_{\pm0.08}$(0.24)	85.08$_{\pm0.07}$(0.42)	84.53$_{\pm0.09}$(0.13)	83.55$_{\pm0.42}$(1.11)	84.91$_{\pm0.15}$(0.25)
	FATrain	80.08$_{\pm0.81}$	93.12$_{\pm0.82}$(13.04)	78.20$_{\pm0.00}$(1.88)	80.88$_{\pm2.57}$(0.80)	80.88$_{\pm2.89}$(0.80)	82.08$_{\pm1.77}$(2.00)	49.68$_{\pm12.93}$(30.40)	81.40$_{\pm1.40}$(1.32)
	FATest	83.60$_{\pm1.62}$	91.40$_{\pm0.49}$(7.80)	71.00$_{\pm0.00}$(12.60)	73.20$_{\pm2.04}$(10.40)	75.40$_{\pm1.36}$(8.20)	73.40$_{\pm1.36}$(10.20)	40.60$_{\pm8.33}$(43.00)	80.20$_{\pm2.14}$(3.40)
	MIA	57.36$_{\pm1.78}$	60.88$_{\pm1.01}$(3.52)	41.40$_{\pm0.00}$(15.96)	0.20$_{\pm0.00}$(57.16)	0.00$_{\pm0.00}$(57.36)	0.20$_{\pm0.00}$(57.16)	1.48$_{\pm0.37}$(55.88)	48.12$_{\pm1.88}$(9.24)
	AvgGap	-	6.21	7.85	17.15	16.70	17.37	32.60	**3.55**

of above metrics, **AvgGap**, since our goal is not to maximize or minimize any single metric, but rather to **reduce the total performance gap** with the retrain model.

3.2 Results

Full-class and sub-class unlearning. Tab. 1 and Tab. 2 show the performance of different methods using ResNet18 model on CIFAR-100/CIFAR-20. As indicated by the **AvgGap** metric, NatMU achieves the smallest average performance gap in two class-unlearning scenarios across six different class-unlearning settings. In full-class unlearning, all MU methods can obtain good performances while NatMU can achieve a smaller performance gap, especially on **MIA** metric. Sub-class unlearning becomes more complex than full-class unlearning. When the forgetting sub-class varies, the behavior of the retrained model on these sub-class data is completely different. However, our approach not only demonstrates good performances but also exhibits remarkable **stability** across various classes.

Table 3: Random-subset unlearning on CIFAR-10 using VGG16-BN and TinyImageNet-200 using ResNet34 under 1% forgetting ratio and 10% forgetting ratio. The results are given by $\boldsymbol{a}_{\pm \boldsymbol{b}}(\boldsymbol{c})$, sharing the same format with Tab. 1.

	Dataset	Metric	Retrain	Finetune	Amnesiac [13]	BadTeacher [14]	SalUn [16]	NatMU
1%	C-10	RA	93.26$_{\pm0.12}$	93.35$_{\pm0.05}$(0.09)	93.01$_{\pm0.05}$(0.24)	93.16$_{\pm0.04}$(0.10)	92.82$_{\pm0.04}$(0.43)	93.01$_{\pm0.09}$(0.24)
		FA	92.80$_{\pm0.80}$	98.72$_{\pm0.48}$(5.92)	92.52$_{\pm0.59}$(0.28)	91.60$_{\pm0.68}$(1.20)	93.00$_{\pm0.52}$(0.20)	93.08$_{\pm0.41}$(0.28)
		MIA	80.16$_{\pm0.73}$	86.88$_{\pm1.26}$(6.72)	45.00$_{\pm0.75}$(35.16)	51.52$_{\pm0.85}$(28.64)	57.06$_{\pm0.88}$(23.10)	71.32$_{\pm0.83}$(8.84)
		AvgGap	-	4.24	11.89	9.98	7.91	**3.12**
	Tiny	RA	66.93$_{\pm0.23}$	66.69$_{\pm0.26}$(0.24)	65.59$_{\pm0.11}$(1.34)	66.35$_{\pm0.10}$(0.59)	65.59$_{\pm0.10}$(1.34)	66.74$_{\pm0.20}$(0.20)
		FA	68.00$_{\pm1.56}$	85.80$_{\pm0.57}$(17.80)	67.30$_{\pm0.35}$(0.70)	70.56$_{\pm0.71}$(2.56)	69.10$_{\pm0.32}$(1.10)	68.68$_{\pm0.72}$(0.68)
		MIA	49.06$_{\pm1.18}$	63.68$_{\pm0.77}$(14.62)	0.90$_{\pm0.09}$(48.16)	0.00$_{\pm0.00}$(49.06)	10.69$_{\pm0.45}$(38.37)	36.76$_{\pm0.55}$(12.30)
		AvgGap	-	10.89	16.73	17.40	13.60	**4.39**
10%	C-10	RA	93.09$_{\pm0.09}$	93.34$_{\pm0.10}$(0.25)	92.79$_{\pm0.07}$(0.29)	92.91$_{\pm0.06}$(0.18)	92.62$_{\pm0.10}$(0.46)	92.16$_{\pm0.07}$(0.93)
		FA	93.64$_{\pm0.20}$	98.74$_{\pm0.06}$(5.09)	93.56$_{\pm0.20}$(0.08)	93.44$_{\pm0.21}$(0.20)	93.15$_{\pm0.18}$(0.50)	93.88$_{\pm0.17}$(0.23)
		MIA	80.59$_{\pm0.43}$	88.66$_{\pm0.18}$(8.06)	36.37$_{\pm0.46}$(44.22)	32.64$_{\pm0.45}$(47.95)	40.13$_{\pm0.55}$(40.46)	73.80$_{\pm0.28}$(6.79)
		AvgGap	-	4.47	14.87	16.11	13.81	**2.65**
	Tiny	RA	65.47$_{\pm0.20}$	66.60$_{\pm0.11}$(1.13)	61.62$_{\pm0.19}$(3.85)	63.54$_{\pm0.08}$(1.93)	60.98$_{\pm0.15}$(4.49)	64.87$_{\pm0.15}$(0.60)
		FA	66.00$_{\pm0.14}$	85.66$_{\pm0.17}$(19.66)	66.65$_{\pm0.33}$(0.65)	66.17$_{\pm0.31}$(0.16)	66.79$_{\pm0.52}$(0.79)	65.72$_{\pm0.33}$(0.28)
		MIA	46.75$_{\pm0.72}$	62.91$_{\pm0.53}$(16.16)	1.25$_{\pm0.05}$(45.50)	0.00$_{\pm0.00}$(46.75)	10.96$_{\pm0.19}$(35.79)	34.66$_{\pm0.35}$(12.09)
		AvgGap	-	12.32	16.66	16.28	13.69	**4.33**

Random-subset unlearning. Tab. 3 demonstrates the performance of different methods in random-subset unlearning. Apart from the CIFAR-10 dataset, NatMU achieves the smallest performance gap across all other datasets and settings. On CIFAR-10 dataset, since the forgetting accuracy and MIA ratio are very high, although *Finetune* cannot guarantee the achievement of unlearning objectives, it achieves the smallest average performance gap. NatMU can also obtain a second smallest performance gap with a much lower gap on **FA** metric. Because the compared three optimization-based methods assign incorrect labels to the forgetting samples, the predictions after the unlearning process is totally different from the remaining samples, resulting in an extremely low **MIA**. It can lead to a "Streisand Effect", where the forgetting samples is actually more noticeable [34].

Moreover, compared to these three methods, NatMU enjoys a significantly smaller **RA** decrease on more complex dataset and larger forgetting ratio, since the conflicts in our fine-tuning data are much fewer.

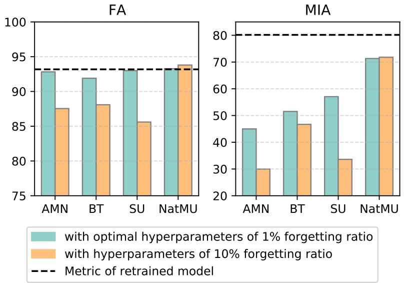


Figure 4: Performance under a forgetting ratio of 1% on CIFAR-100, with the optimal hyperparameters and hyperparameters from a forgeting ratio of 10%. AMN: Amnesiac, BT: BadTeacher, SU:SalUn.

Performance with transferred hyperparameters. A possible solution for the absence of the retrained model as the hyperparameter tuning reference is to identify a set of hyperparameters under a specific experimental settings and transfer them to other settings. Fig. 4 illustrates the performance of different methods when transferring the optimal hyperparameters with 10% forgetting ratio to 1%. Due to the existence of remaining data, **RA** of four methods remains almost unchanged, thus not reported here. In case of **FA** and **MIA**, NatMU exhibits a stable performance compared to other baselines, making it a promising candidiate for pratical MU.

4 Conclusion

Revisiting current popular machine unlearning methods, we identify their unnatural properties and resulting issues, such as unnatural generalization and impracticality. To address these issues, we propose a straightforward but effective machine unlearning method, NatMU, which injects correct information from remaining data into forgetting samples to achieve unlearning. NatMU demonstrates superior performance and robustness across different unlearning settings. Our initial step towards natural machine unlearning opens up new perspectives for achieving more efficient and effective MU.

Limitations and future work. Firstly, although NatMU performs well, it lacks theoretical support, which means it is not certified. This weakness is also shared by the compared baseline methods. Secondly, the weighting vectors are currently designed manually; thus, there is potential for improving, which we leave for future works. Thirdly, our algorithm needs to access the whole remaining data. Enhancements in efficiency could be achieved through methods such as dataset distillation [35].

References

[1] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, pages 10073–10082. Computer Vision Foundation / IEEE, 2020.

[2] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: A survey. *ACM Comput. Surv.*, 55(5):97:1–97:37, 2023.

[3] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[4] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015.

[5] Chris Jay Hoofnagle, Bart Van Der Sloot, and Frederik Zuiderveen Borgesius. The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*, 28(1):65–98, 2019.

[6] Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In *NeurIPS*, pages 3513–3526, 2019.

[7] Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 3832–3842. PMLR, 2020.

[8] Enayat Ullah, Tung Mai, Anup Rao, Ryan A. Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *COLT*, volume 134 of *Proceedings of Machine Learning Research*, pages 4126–4142. PMLR, 2021.

[9] Goel Shashwat, Prabhu Ameya, Sanyal Amartya, Lim Ser-Nam, Torr Philip, and Kumaraguru Ponnurangam. Towards adversarial evaluations for inexact machine unlearning. *CoRR*, abs/2201.06640, 2022.

[10] Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan S. Kankanhalli. Zero-shot machine unlearning. *IEEE Trans. Inf. Forensics Secur.*, 18:2345–2354, 2023.

[11] Stefan Schoepf, Jack Foster, and Alexandra Brintrup. Parameter-tuning-free data entry error unlearning with adaptive selective synaptic dampening. *CoRR*, abs/2402.10098, 2024.

[12] Ga Wu, Masoud Hashemi, and Christopher Srinivasa. PUMA: performance unchanged model augmentation for training data removal. In *AAAI*, pages 8675–8682. AAAI Press, 2022.

[13] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI*, pages 11516–11524. AAAI Press, 2021.

[14] 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. In *AAAI*, pages 7210–7217. AAAI Press, 2023.

[15] Pratyush Maini, Michael Curtis Mozer, Hanie Sedghi, Zachary Chase Lipton, J. Zico Kolter, and Chiyuan Zhang. Can neural network memorization be localized? In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 23536–23557. PMLR, 2023.

[16] 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. *CoRR*, abs/2310.12508, 2023.

[17] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *USENIX Security Symposium*, pages 4007–4022. USENIX Association, 2022.

[18] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling SGD: understanding factors influencing machine unlearning. In *EuroS&P*, pages 303–319. IEEE, 2022.

[19] Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *CoRR*, abs/2308.07061, 2023.

[20] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*. OpenReview.net, 2017.

[21] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR (Poster)*. OpenReview.net, 2018.

[22] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6022–6031. IEEE, 2019.

[23] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *ICANN (2)*, volume 9887 of *Lecture Notes in Computer Science*, pages 63–71. Springer, 2016.

[24] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

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

[26] Mohammed Ali mnmoustafa. Tiny imagenet, 2017.

[27] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *AAAI*, pages 12043–12051. AAAI Press, 2024.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.

[30] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. In *NDSS*. The Internet Society, 2023.

[31] Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. In *NeurIPS*, 2023.

[32] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *NeurIPS*, 2023.

[33] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*, pages 2615–2632. USENIX Association, 2021.

[34] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *CVPR*, pages 9301–9309. Computer Vision Foundation / IEEE, 2020.

[35] Junaid Iqbal Khan. Dataset condensation driven machine unlearning. *CoRR*, abs/2402.00195, 2024.

[36] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy*, pages 463–480. IEEE Computer Society, 2015.

[37] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 2017.

[38] Ryutaro Tanno, Melanie F. Pradier, Aditya V. Nori, and Yingzhen Li. Repairing neural networks by leaving the right past behind. In *NeurIPS*, 2022.

[39] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *AISTATS*, volume 130 of *Proceedings of Machine Learning Research*, pages 2008–2016. PMLR, 2021.

[40] Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N. Ravi. Deep unlearning via randomized conditionally independent hessians. In *CVPR*, pages 10412–10421. IEEE, 2022.

[41] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. In *NDSS*. The Internet Society, 2023.

[42] James Martens. New insights and perspectives on the natural gradient method. *J. Mach. Learn. Res.*, 21:146:1–146:76, 2020.

[43] Xinwen Cheng, Zhehao Huang, and Xiaolin Huang. Machine unlearning by suppressing sample contribution. *CoRR*, abs/2402.15109, 2024.

[44] Jack Foster, Kyle Fogarty, Stefan Schoepf, Cengiz Öztireli, and Alexandra Brintrup. Zero-shot machine unlearning at scale via lipschitz regularization. *CoRR*, abs/2402.01401, 2024.

[45] Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *AAAI*, pages 11186–11194. AAAI Press, 2024.

Related Work

Machine unlearning aims to update a pre-trained model to remove the influence of a subset of training set [36]. Although retraining from scratch with the remaining data can achieve perfect MU, its significant demand for resources and time is unacceptable in the current deep learning context. Therefore, numerous works have emerged that aim to approximate the retrained model, especially in the output space.

Optimization-free unlearning focuses on removing the influence of the forgetting data by directly modifying model weights. Influence function is first introduced by [37], and [7] adopts it for certified data removal in L_2-regularized linear models. The following work [38, 39, 40, 41, 42, 27] is dedicated to reducing the computational burdens introduced by Hessian inversion. [39] adopts infinitesimal jackknife with Newton methods to reduce the number of calculations while [40] reduces the overhead of one calculation by selecting only important parameters. FisherForgetting [34] introduces a weight scrubbing method by injecting noise to specific parameters to clean the information about the forgetting data with fisher information matrix [42]. SSD [27] addresses its computational overhead and generalization decrease through a fast but stringent parameter selection. Besides the influence function-based methods, [13] proposes to delete the gradient of mini-batches relevant to the forgetting data. [18] decomposes the SGD training process and performs a gradient ascent on the forgetting data. Although these methods can remove the influence of forgetting data, they may significantly impair the generalization capability of the MU model, especially DNN.

Optimization-based unlearning re-optimizes the original model on a carefully crafted dataset with a proposed unlearning objective. Amnesiac [13] fine-tunes the model with randomly labeled forgetting samples along with unchanged remaining data. BadTeacher [14] and SCRUB [32] both achieve machine unlearning under a teacher-student framework, where the student network selectively obeys the teacher models for different samples. SalUn [16] adopts the same way as Amnesiac, while constraining the model parameters' update with a weight saliency mask. These methods relabel the forgetting data to some extent. Another line of works adopts gradient ascent on forgetting data to achieve unlearning, combined with gradient descent on remaining data to maintain model performance. They also face the problem that how to determine the stop point of the unlearning process. Recently, some studies have explored the scenario where no remaining data are available, called "zero-shot" machine unlearning. [] generates adversarial samples from forgetting samples and ensures that the predictions of the MU model on these samples match those of the original model, thereby maintaining model performance in the absence of remaining data. MU-Mis [43] reveals the link between sample's contribution to learning process and model's sensitivity to it, subsequently proposing an algorithm by minimizing input sensitivity. JiT [44] and [45] perturbs forgetting samples with random or adversarial perturbation, and ensures the predictions of perturbed versions match reference predictions to maintain model performance.