Partially Blinded Unlearning: Class Unlearning for Deep Networks from Bayesian Perspective

Subhodip Panda[1], **Shashwat Sourav**[2], **Prathosh A.P.**[1]
[1]Indian Institute of Science, Bangalore
[2]Indian Institute of Science Education and Research, Bhopal
subhodipp@iisc.ac.in, shashwat20@iiserb.ac.in, prathosh@iisc.ac.in

Abstract

To follow regulations on individual data privacy and safety, machine learning models must systematically remove information learned from specific subsets of a user's training data that can no longer be utilized. To address this problem, *machine unlearning* has emerged as an important area of research, that helps remove information learned from specific subsets of training data from a pre-trained model without needing to retrain the whole model from scratch. The principal aim of this study is to formulate a methodology aimed for the purposeful elimination of information linked to a specific class of data from a pre-trained classification network. This intentional removal decreases the model's performance specifically concerning the unlearned data class while simultaneously minimizing any detrimental impacts on the model's performance in other classes. To achieve this goal, we frame the class unlearning problem from a Bayesian perspective, which yields a loss function that minimizes the log-likelihood associated with the unlearned data with a stability regularization in parameter space. This stability regularization incorporates Mohalanobis distance with respect to the Fisher Information matrix and l_2 distance from the pre-trained model parameters. Our novel approach, termed *Partially-Blinded Unlearning (PBU)*, surpasses existing state-of-the-art class unlearning methods, demonstrating superior effectiveness. Notably, *PBU* achieves this efficacy without requiring information about the entire training dataset but only of the unlearned data points, marking a distinctive feature of its performance.

Code — https://github.com/Subhodip123/pbu

Introduction

In recent years, there has been a surge in the extensive training of machine learning models, utilizing large volumes of user data across diverse applications. As a result, users' public and private data has become an integral part of these models. This development prompts a critical inquiry:

How can individuals control the utilization of their personal data used by these models?

To address this question, contemporary regulatory frameworks concerning data privacy and protection, exemplified by the California Consumer Privacy Act (CCPA) (Goldman 2020) and the European Union's General Data Protection Regulation (GDPR) (Voigt and dem Bussche 2017), mandate organizations to implement rigorous controls on models. Specifically, these regulations require organizations not only to delete personal information from databases but also to remove corresponding learned information from the trained models upon the users' request. One naive approach to tackle this issue involves fully retraining a machine-learning model using the remaining data after removing the data intended for unlearning. Nevertheless, this methodology proves impractical, given its significant demands on time and space resources, especially when dealing with inaccessible remaining datasets, a common occurrence in practical scenarios. Hence, the process of unlearning persists as a challenging endeavor.

The framework of Machine Unlearning (Xu et al. 2020; Nguyen et al. 2022b) tries to tackle the above-mentioned challenges. Specifically, machine unlearning pertains to the task of either forgetting the learned information (Sekhari et al. 2021; Ma et al. 2022; Ye et al. 2022; Cao and Yang 2015; Golatkar et al. 2021a; Golatkar, Achille, and Soatto 2020c; Ginart et al. 2019; Golatkar, Achille, and Soatto 2020a) or erasing the influence (Wu, Dobriban, and Davidson 2020; Guo et al. 2020; Graves, Nagisetty, and Ganesh 2021a; Wu, Hashemi, and Srinivasa 2022; Wu, Tannen, and Davidson 2020; Chourasia and Shah 2023) of a specific subset of training dataset from a learned model, in response to a user's request. Current methodologies for machine unlearning within classification networks primarily revolve around the endeavor of unlearning a small subset or the entirety of data points associated with a specific class, whether it be singular or multiple classes. This specific process is commonly referred to as *class unlearning.* This task poses an inherent challenge as it requires the targeted unlearning of a particular class within the training data while avoiding adverse effects on the previously acquired knowledge from other classes of data points. In essence, the unlearning process introduces the risk of *catastrophic forgetting*, as evidenced by prior studies (Ginart et al. 2019; Nguyen et al. 2022a; Golatkar, Achille, and Soatto 2020a). This risk may lead to a significant decrease in the overall performance of the model, particularly in classes other than the one targeted for unlearning. To address these challenges, current meth-

ods (Chundawat et al. 2023; Tarun et al. 2023) in class unlearning assume access to the entire dataset and employ two-step strategies to restore the model's performance on other data classes. In some cases (Wu, Dobriban, and Davidson 2020; Graves, Nagisetty, and Ganesh 2021a), these methods rely on stringent assumptions over the training procedure only applicable to small models.

Our study addresses the challenge of class unlearning from a pre-trained classification network within a stricter scenario where the other class data is inaccessible. Inspired by the work of (Nguyen, Low, and Jaillet 2020), we adopt a Bayesian perspective, framing the problem as the unlearning of specific data points. The theoretical formulation yields a loss function with the objective of minimizing the log-likelihood associated with the unlearned class data, incorporating a stability regularization term. Utilizing the second-order Taylor approximation and the assumption of parameter Gaussianity, the stability regularization can be decomposed into two main components. The first component centers on the Mahalanobis distance, computed relative to the Fisher Information matrix of the initial parameters. Meanwhile, the second component relates to the l_2 distance, quantifying the disparity between the unlearned model and the initial model. Notably, this formulation allows for a natural interpretation of a trade-off: fully unlearning the target data class versus retaining some knowledge of other class data encoded in the initial parameters. The latter mitigates the risk of catastrophic unlearning induced by the former. Furthermore, our approach distinguishes itself by not demanding access to the complete dataset, a requirement prevalent in many contemporary methods (Tarun et al. 2023; Chundawat et al. 2023; Graves, Nagisetty, and Ganesh 2021b). Instead, it only requires access to the unlearned data to yield results in a one-step method. In essence, it remains oblivious to the retaining data classes, earning its nomenclature as Partially-Blinded Unlearning. Our contributions can be summarized as follows:

- We provide a theoretical formulation for Class Unlearning and propose a methodology applicable to unlearning specific classes in deep neural networks.
- Our method exhibits superiority over concurrent class unlearning methods (Chundawat et al. 2023; Tarun et al. 2023) by not mandating access to the entire dataset, only requiring partial access to the unlearned data. This feature proves advantageous in more restricted experimental settings.
- Our method's single-step unlearning process contrasts with two-step approaches used by some contemporary methods (Tarun et al. 2023), showcasing superior computational efficiency and simplicity
- We validate our approach using ResNet-18, ResNet-34, ResNet-50, Densenet-121, All-CNN, and ConvNeXt-Large models across three vision datasets: MNIST, CIFAR-100 and Food101, demonstrating its generalizability across different models and datasets. Our method consistently outperforms many state-of-the-art class unlearning methods.

Related Works

Machine Unlearning

Machine unlearning, as defined by the literature (Xu et al. 2020; Nguyen et al. 2022b), involves intentionally removing specific acquired knowledge or erasing the impact of particular subsets of training data from a trained model. The naive approach to unlearning is to remove the undesired data subset from the training dataset and then retrain the model from scratch with the rest of the training data. However, this method becomes computationally expensive due to the large volume of training data.

Subsequently, inspired by user privacy concerns, Cao and Yang (2015) developed efficient methods for deleting data from certain statistical query algorithms, coining the term *machine unlearning*. Unfortunately, these techniques are primarily suited for structured problems and do not extend well to complex machine learning algorithms like k-means clustering or random forests (Brophy and Lowd 2021). Later, unlearning algorithms were proposed for the k-means clustering problem (Ginart et al. 2019), introducing effective data unlearning criteria applicable to randomized algorithms based on statistical indistinguishability. Building upon this criterion, machine unlearning methods are broadly categorized into two main types: exact unlearning (Ginart et al. 2019; Brophy and Lowd 2021) and approximate unlearning (Neel, Roth, and Sharifi-Malvajerdi 2021). Exact unlearning aims to completely remove the effect of unwanted data from the trained model. This requires the parameters of the unlearned model to exactly match those of a retrained model. In contrast, approximate unlearning methods only partly remove the data's influence, resulting in parameter distributions closely resembling the retrained model's. To achieve exact unlearning, Wu et. al. (Wu, Dobriban, and Davidson 2020) proposed a technique employing subtracting the cached gradients of the unlearning data, offering computational efficiency while increasing memory usage. Also, this method needs awareness of the unlearning data during the training phase, which is quite restrictive. More sophisticated methods (Guo et al. 2020; Graves, Nagisetty, and Ganesh 2021a) have suggested using influence functions for this purpose. However, these approaches are computationally demanding due to the requirement of Hessian inversion techniques and are limited to small convex models.

To broaden the applicability of unlearning techniques to non-convex models such as deep neural networks, (Golatkar, Achille, and Soatto 2020a) introduced a scrubbing mechanism that uses the Fisher Information matrix. This mechanism is designed for unlearning a specific class of data, focusing on the task of class unlearning. Despite the possibility of achieving zero accuracy on unlearned class data, (Tarun et al. 2023) demonstrates that this scrubbing mechanism often leads to poor accuracy on data from retaining classes. For further improvement, (Chundawat et al. 2023) introduced a method based on the knowledge distillation approach, employing two teacher networks. One is identified as a proficient teacher, trained on the entire dataset, while the other is an unskilled teacher initialized randomly. The methodology

involves fine-tuning a student network and adjusting its parameters so that its output aligns with the proficient teacher for the retaining data class and the unskilled teacher for the unlearning data class. Despite achieving state-of-the-art results, this method requires access to the complete training dataset, posing a significant restriction. Our current method (PBU) is close to the works of (Golatkar, Achille, and Soatto 2020a) as it also uses the Fisher Information Matrix of the initial parameters but produces superior to comparable performance.

Continual Learning

Continual learning constitutes an active area of research, to develop methods that enable the model to adapt to future tasks while preserving its performance on previously acquired tasks. Even though the objective of continual learning and machine unlearning is orthogonal, many techniques of continual learning can be applied to the task of unlearning (Liu, Liu, and Stone 2022; Tanno et al. 2022). Especially the task of incrementally learning using the Elastic Weight Consolidation (EWC) regularization (Kirkpatrick et al. 2017; Schwarz et al. 2018) is closely related to our method of unlearning. If the Fisher Information matrix becomes diagonal for our case the stability regularizer resolves to EWC regularization technique. Additionally, one has the flexibility to explore enhancements beyond EWC, such as online EWC (Schwarz et al. 2018), or other regularization-based techniques grounded in Bayesian learning principles (Nguyen et al. 2017; Pan et al. 2020; Loo, Swaroop, and Turner 2021).

Methodology

Unlearning Problem Formulation

Consider a particular parameter in the parameter space $\Theta \subseteq \mathbb{R}^m$ is denoted as θ. Now the pre-trained classification model denoted as f_{θ^*} with initial parameters $\theta^* \in \Theta$. This classifier undergoes training using a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, where $(x_i, y_i) \overset{iid}{\sim} P_{XY}(x, y)$. Here x_i signifies the feature vector belonging to the feature space $\mathcal{X} \subseteq \mathbb{R}^d$, and y_i represents the corresponding label or class belonging to the label space $\mathcal{Y} = \{0, 1, 2, \ldots, C-1\}$, where C stands for the total number of classes in the dataset. In response to a user-specified request, indicating a particular class or classes of data points $s_n \in \mathcal{Y}$ that the model needs to unlearn, we gain access to the unlearning samples of that specific class denoted as $\mathcal{S}_n = \{(x_i, y_i) : y_i = s_n\}$. The objective of unlearning is to determine a parameter θ^u for the unlearned model f_{θ^u} that closely aligns with the performance of the retrained model f_{θ^p}, which is trained on samples $\mathcal{S}_p = \mathcal{D} \setminus \mathcal{S}_n$. Now, given that the retrained model achieves low accuracy on unlearned class samples and high accuracy on retaining class samples, the task involves adjusting the initial model parameter θ^* to θ^u such that the accuracy of f_{θ^u} on unlearned class samples $\mathcal{S}_n$ decrease, while maintaining similar performance as the retrained model on retained samples $\mathcal{S}_p$. It's crucial to highlight that we work under a more restrictive condition where access to $\mathcal{S}_p$ is prohibited, rendering us unaware of the entire dataset $\mathcal{D} = \mathcal{S}_p \cup \mathcal{S}_n$. Under this constraint of restricted access, retraining the model is infeasible in our scenario.

Unlearning Problem: Bayesian view

Given θ^* and $\mathcal{S}_n$, the unlearning objective is to find the unlearned parameter θ^u that closely resembles the retrained parameter θ^p expressed via a maximum a posteriori estimate given by $\theta^p = \arg\max_\theta P(\theta|\mathcal{S}_p)$, which is expanded as below:

$$\theta^p = \arg\max_\theta \log P(\theta|\mathcal{S}_p) \tag{1}$$

$$= \arg\max_\theta \log P(\mathcal{S}_p|\theta) + \log P(\theta) - \log P(\mathcal{S}_p) \tag{2}$$

$$= \arg\max_\theta \log P(\mathcal{S}_p|\theta) + \log P(\theta) - K_1 \tag{3}$$

Eq. 2 is obtained via Bayes Rule and the term $\log P(\mathcal{S}_p)$ is replaced by a constant K_1 since it is independent of θ, to obtain Eq. 3. Now from the pre-trained model on the whole dataset $\mathcal{D}$, we have the following:

$$\log P(\theta|\mathcal{D}) = \log P(\theta|\mathcal{S}_p, \mathcal{S}_n) \tag{4}$$

$$= \log P(\mathcal{S}_p, \mathcal{S}_n|\theta) + \log P(\theta) - K_2 \tag{5}$$

$$= \log P(\mathcal{S}_p|\theta) + \log P(\mathcal{S}_n|\theta) + \log P(\theta) - K_2 \tag{6}$$

Now substituting the value of $\log P(\mathcal{S}_p|\theta) + \log P(\theta)$ from Eq. 6 to Eq. 3, we get the following:

$$\theta^p = \arg\max_\theta \log P(\theta|\mathcal{D}) - \log P(\mathcal{S}_n|\theta) + K_2 - K_1 \tag{7}$$

$$\equiv \arg\max_\theta \log P(\theta|\mathcal{D}) - \log P(\mathcal{S}_n|\theta) \tag{8}$$

$$= \arg\max_\theta \mathcal{L}(\theta, \mathcal{D}, \mathcal{S}_n) \tag{9}$$

Proposed Method: Overview

The task of finding the unlearned parameter θ^u, that approximates θ^p in Eq. 9 demands access to the complete dataset $\mathcal{D}$, which is infeasible. Given that only the unlearning data $\mathcal{S}_n$ is accessible, we reformulate the objective (maximizing $\mathcal{L}(\theta, \mathcal{D}, \mathcal{S}_n)$) by constructing an upper bound on it[1]. This results in the minimization of a loss function $\mathcal{L}(\theta, \theta^*, \mathcal{S}_n)$ (shown in Remark 1), as given in Eq. 10.

$$\begin{aligned}\mathcal{L}(\theta, \theta^*, \mathcal{S}_n) = \alpha \log P(\mathcal{S}_n|\theta) \\ + \beta(\theta - \theta^*)^T I_{\theta^*}(\mathcal{S}_n)(\theta - \theta^*) \\ + \gamma||\theta - \theta^*||^2\end{aligned} \tag{10}$$

Here, α, β, γ are hyper-parameters and $I_{\theta^*}(\mathcal{S}_n)$ is the Fisher Information Matrix of the initial parameters corresponding to the unlearning data class. The overall methodology, illustrated in Figure 1, entails perturbing the initial parameter θ^* to the unlearned parameter θ^u using the loss function described in Eq. 10. The loss function comprises

[1]While maximization of a lower bound of the objective is desirable, we resort to an upper bound optimization since construction of a lower bound is intractable in the current problem and show that it leads to better performance empirically.

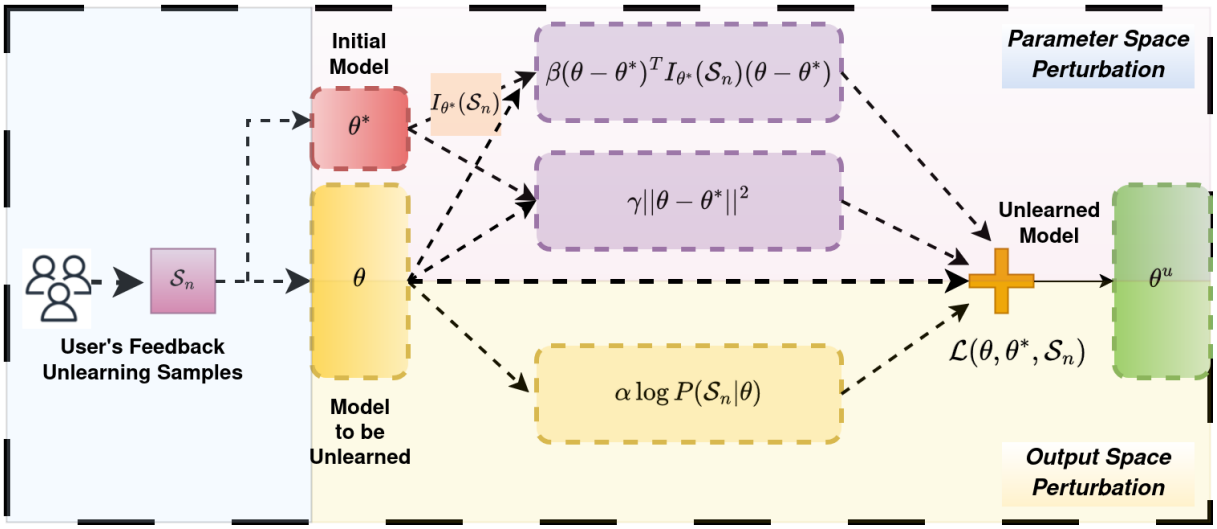

Figure 1: **Partially Blinded Unlearning (PBU) Method:** Given user-identified samples to be unlearned ($\mathcal{S}_n$), our unlearning method employs a two-component perturbation technique, indicated by a loss function comprising three terms: the first term(shown in the bottom half) represents the perturbation in the output space, aiming to minimize the log-likelihood associated with the unlearned class while the last two terms correspond to perturbations in the parameter space (shown in the upper half), including the Mahalanobis Distance with respect to the Fisher Information matrix and the l_2 distance.

Algorithm 1: Partially Blinded Unlearning (PBU)

Input: Unlearning Data Classes: $\mathcal{S}_n$, Initial Paramter: θ^*, hyper-parameters: α, β, γ

Initialize: $\theta \leftarrow \theta^*$

Calculate Fisher Information Matrix: $I_{\theta^*}(\mathcal{S}_n)$

for $t \leq T_{ul}$ **do**

Calculate $\mathcal{L}(\theta,\theta^*,\mathcal{S}_n) = \alpha \log P(\mathcal{S}_n|\theta) + \beta(\theta-\theta^*)^T I_{\theta^*}(\mathcal{S}_n)(\theta-\theta^*) + \gamma||\theta-\theta^*||^2$

$\theta^{t+1} \leftarrow \theta^t - \eta \nabla_\theta \mathcal{L}(\theta,\theta^*,\mathcal{S}_n)$

end for

Output: $\theta^{T_{ul}}$

two components: the first component involves perturbing the parameters by minimizing the log-likelihood associated with the unlearning class data, while the second component involves perturbing parameters by incorporating stability regularization in the parameter space. This stability regularization consists of the Mahalanobis distance between the initial parameter and the unlearned parameter with respect to the Fisher Information matrix corresponding to the negative class, along with the l_2 distance between the initial parameter and the unlearned parameter. Through this methodology, our method aims to strike a balance between erasing specific class information and retaining overall model performance and robustness. In our formulation, the dependence on $\mathcal{D}$ is encapsulated in the initial parameter θ^*. This characteristic provides an advantage to our method, as it only requires access to the unlearning class data points, making it ***'partially blind'*** to the whole dataset. Also, our method (*PBU*) in Algorithm 1 is a single-step method and does not require repairing as suggested in many of the algorithms (Tarun et al. 2023; Chundawat et al. 2023). In the subsequent section, we provide a theoretical explanation detailing the construction of the loss function utilized in our methodology.

Construction of Loss Function: Theoretical Outlook

Definition 1 ***(Fisher Information Matrix):*** *$\forall\theta \in \Theta$ the Fisher Information Matrix ($I_\theta(\mathcal{D})$) for the model on the whole dataset $\mathcal{D}$ is defined as follows:*

$$I_\theta(\mathcal{D}) = \mathbb{E}_{P_\theta(\mathcal{D})}\left[\nabla_\theta \log P_\theta(\mathcal{D}) \nabla_\theta \log P_\theta(\mathcal{D})^T\right] \quad (11)$$

Regularity Conditions: Now the above definition can further be simplified if we assume certain regularity conditions as follows:

1. Θ be an open set in $\mathbb{R}^m$
2. The partial derivatives $\frac{\partial P_\theta(\mathcal{D})}{\partial \theta_i}$ and $\frac{\partial^2 P_\theta(\mathcal{D})}{\partial \theta_i^2}$; $\forall i \in [m]$ exists and is finite $\forall\theta \in \Theta$ and $\forall x \in \mathcal{X}$
3. $\int_{\mathcal{X}} \nabla_\theta P_\theta(\mathcal{D}) d\mu(\mathcal{D}) = \nabla_\theta \int_{\mathcal{X}} P_\theta(\mathcal{D}) d\mu(\mathcal{D}) = 0$
4. $\int_{\mathcal{X}} \nabla^2_\theta P_\theta(\mathcal{D}) d\mu(\mathcal{D}) = \nabla^2_\theta \int_{\mathcal{X}} P_\theta(\mathcal{D}) d\mu(\mathcal{D}) = 0$

Theorem 1 (Moulin and Veeravalli (2019) Lemma-13.1) *If the regularity conditions (1)-(4) hold then $\forall\theta \in \Theta$ the Fisher Information matrix can be written in the following form*

$$I_\theta(\mathcal{D}) = \mathbb{E}_{P_\theta(\mathcal{D})}\left[-\nabla^2_\theta \log P_\theta(\mathcal{D})\right] \quad (12)$$

Lemma 1 *If the regularity conditions (1)-(4) hold then for the pretrained initial parameter θ^*, training data D and unlearned class $\mathcal{S}_n$*

$$I_{\theta^*}(\mathcal{D}) \geq I_{\theta^*}(\mathcal{S}_n) \quad (13)$$

The proof of Theorem-1 and Lemma-1 are included in the appendix A.1 and A.2 respectively. Lemma-1 uses, the i.i.d assumption, $\forall\theta \in \Theta, P_\theta(\mathcal{S}_p, \mathcal{S}_n) = P_\theta(\mathcal{S}_p)P_\theta(\mathcal{S}_n)$. and linearity of expectation and marginalization.

Lemma 2 *(Koh and Liang 2017; Wu, Hashemi, and Srinivasa 2022; Tanno et al. 2022) If θ^* is the MAP estimate of $P(\theta|\mathcal{D})$ then for some constant K_3 independent of the parameter θ the following approximation holds.*

$$\log P(\theta|\mathcal{D}) \approx \frac{1}{2}(\theta-\theta^*)^T \nabla^2_\theta \log P(\theta^*|\mathcal{D})(\theta-\theta^*) + K_3 \tag{14}$$

Theorem 2 *If the above regularity conditions (1)-(4) hold and if the pre-trained initial parameter θ^* is the MAP estimate of $P(\theta|\mathcal{D})$ with prior parameter distribution $P(\theta) = \mathcal{N}(0, \lambda I)$ with some variance parameter $\lambda \geq 0$ then*

$$\begin{aligned}\mathcal{L}(\theta, \mathcal{D}, \mathcal{S}_n) \leq &- \log P(\mathcal{S}_n|\theta) \\ &- \frac{N}{2}(\theta-\theta^*)^T I_{\theta^*}(\mathcal{S}_n)(\theta-\theta^*) - \frac{\lambda}{2}||\theta-\theta^*||^2 + K_3\end{aligned} \tag{15}$$

The proof of Lemma-2 and Theorem-2 are included in Appendix A.3 and A.4, respectively.

Remark 1 *Now using above Theorem-2 in Eq. 9 our final objective to find θ^p results in the desired loss function of Eq. 10 as follows*

$$\begin{aligned}\theta^p &= \arg\max_\theta \mathcal{L}(\theta, \mathcal{D}, \mathcal{S}_n) \\ &\leq \arg\max_\theta \Big[- \log P(\mathcal{S}_n|\theta) \\ &\quad - \frac{N}{2}(\theta-\theta^*)^T I_{\theta^*}(\mathcal{S}_n)(\theta-\theta^*) - \frac{\lambda}{2}||\theta-\theta^*||^2 + K_3\Big] \\ &\equiv \arg\min_\theta \Big[\log P(\mathcal{S}_n|\theta) \\ &\quad + \frac{N}{2}(\theta-\theta^*)^T I_{\theta^*}(\mathcal{S}_n)(\theta-\theta^*) + \frac{\lambda}{2}||\theta-\theta^*||^2\Big] \\ &\approx \arg\min_\theta \Big[\alpha \log P(\mathcal{S}_n|\theta) \\ &\quad + \beta(\theta-\theta^*)^T I_{\theta^*}(\mathcal{S}_n)(\theta-\theta^*) + \gamma||\theta-\theta^*||^2\Big]\end{aligned}$$

Experiments and Results

Datasets and Models

For class unlearning objective, we have used various classification models such as ResNet18, ResNet34, ResNet50 (He et al. 2016), AllCNN (Challa, Yellamraju, and Bhatt 2019), DenseNet-121 (He et al. 2016), and ConvNeXt-Large (Liu et al. 2022) models for learning and unlearning tasks. Further to evaluate the applicability of our method we take three datasets, i.e., MNIST (LeCun et al. 1998), CIFAR100 (Krizhevsky, Hinton et al. 2009), and Food101 (Bossard, Guillaumin, and Van Gool 2014) dataset. To show the efficacy of our method for different class unlearning settings, we have used different combinations of classes for different datasets and models which can be seen in Table 1. Due to space constraints additional experimental results on Resnet-18, Resnet-34, and All-CNN models are included in Appendix B.1, along with additional results demonstrating the effectiveness of our method on CIFAR-10 in Appendix B.2.

Implementation Details

Initial Training, Unlearning, and Baselines: Initially, we trained all models (referred to as the initial model) using the entire dataset. Subsequently, we partitioned the test data into two segments: one containing the class targeted for unlearning and the other encompassing the remaining classes. We assessed the accuracy of the initial model on both partitions of the test data. In our approach, we solely train the unlearned model using the unlearning class and employ a proposed loss function in Eq. 10 to facilitate unlearning of the desired class. Additional specifics regarding the experimental settings of initial training and unlearning methodologies are provided in Appendix Section C. To evaluate our method, we included four baseline techniques. These are as follows:

- **Retraining:** We trained the model again using data subset $\mathcal{S}_p$. The retrained model exhibited 0% accuracy on the unlearned class $\mathcal{S}_n$, while maintaining similar accuracy on the retained classes $\mathcal{S}_p$.
- **Fine-Tuning:** Rather than retraining from scratch, we initialized the model from the initial pre-trained checkpoint and fine-tuned it using $\mathcal{S}_p$.
- **Fast-Effective:** We conducted a comparative analysis between our method and the *Fast-Yet-Effective* approach proposed by (Tarun et al. 2023). In this method, the authors maximize noise by employing a noise matrix to manipulate the weights of the initial model for unlearning through impair-repair steps, where weights are manipulated in the impair step and stabilized on the retained classes in the subsequent repair step. This required access to the whole dataset $\mathcal{D}$.
- **Bad Teaching:** Additionally, our method was compared with the bad teaching approach proposed by (Chundawat et al. 2023). In this method, a competent teacher, an incompetent teacher, and a student model are involved, where the student model attempts to unlearn classes specified by the user based on the information received from both teachers. Like the previous baseline, it requires access to the whole dataset $\mathcal{D}$ as well.

Evaluation Metrics

In previous works (Golatkar, Achille, and Soatto 2020b,d; Golatkar et al. 2021b; Graves, Nagisetty, and Ganesh 2021b), the following unlearning metrics have been introduced to measure the performance of the unlearning algorithm. These metrics have been developed to measure the amount of information left in the network corresponding to the forget class. We mainly use the following metrics:

- **Accuracy on forget and retain class:** Following unlearning, the model's accuracy on the forgotten class, denoted as A_{D_f}, is expected to be significantly low (approaching zero), while the accuracy on the remaining classes, represented as A_{D_r}, ideally should closely resemble the performance of the retrained model.
- **Membership inference attack (MIA) accuracy:** This metric signifies the model's resilience against membership inference attacks, aiming to prevent the extraction

Dataset	Models	Classes	Initial Training		Re-training		Fine-tuning		Fast-Effective		Bad Teaching		PBU (Our Method)	
			A_{D_f}	A_{D_r}	A_{D_f}	A_{D_r}	A_{D_f}	A_{D_r}	A_{D_f}	A_{D_r}	A_{D_f}	A_{D_r}	A_{D_f}	A_{D_r}
MNIST	Resnet-34	Class-2	99.65±0.11	99.42±0.11	0±0	99.33±0.11	0±0	99.37±0.06	0±0	94.17±0.88	0±0	97.96±0.29	**0.03±0.06**	**98.73±0.57**
		Class-6	98.89±0.59	99.51±0.06	0±0	99.34±0.14	0±0	99.44±0.16	0±0	89.13±2.86	0±0	87.62±0.49	**0±0**	**91.09±2.9**
		Class-8	99.73±0.21	99.41±0.12	0±0	99.31±0.08	0±0	99.37±0.19	0±0	94.64±1.97	0±0	96.18±0.53	**0±0**	**98.24±0.36**
	Densenet-121	Class-2	99.6±0.12	99.44±0.11	0±0	99.15±0.05	0±0	99.37±0.12	0±0	91.43±1.62	0±0	94.15±0.54	**0±0**	**96.5±0.3**
		Class-6	99.72±0.12	99.53±0.1	0±0	99.29±0.05	0±0	99.69±0.16	0±0	96.1±0.65	0±0	97.97±0.23	**0.84±0.35**	**98.65±0.11**
		Class-8	98.95±0.63	99.58±0.11	0±0	99.47±0.09	0±0	99.53±0.08	0±0	94.5±2.34	0±0	94.83±0.59	**0.27±0.24**	**98.57±0.31**
	ConvNeXt-L	Class-2	99.69±0.21	99.55±0.1	0±0	99.11±0.1	0±0	99.19±0.12	0.18±0	97.52±0.24	0±0	97.18±1.14	**0±0**	**98.65±0.29**
		Class-6	99.05±0.05	99.59±0.1	0±0	98.83±0.17	0±0	98.63±0.01	0±0	98.25±0.13	0±0	97.75±0.51	1±0.16	**98.33±0.18**
		Class-8	99.56±0.06	99.6±0.14	0±0	98.81±0.06	0±0	98.85±0.04	0±0	97.54±0.99	0±0	96.51±1.5	1.22±1.56	**97.26±1.59**
CIFAR-100	Resnet-50	Class-1	86.33±7.23	75.87±0.31	0±0	75.2±0.34	0±0	69.98±1.24	0±0	54.61±0.21	0.87±0.23	68.06±0.14	**0±1.15**	**70.51±0.18**
		Class-3	56.67±11.72	76.17±0.41	0±0	74.12±0.93	0±0	69.25±0.36	0±0	59.43±0.56	0±0	70.35±1.19	**0.5±0.58**	**71.85±0.91**
		Class-8	92.33±2.89	75.81±0.34	0±0	74.31±0.83	0±0	68.28±0.66	0±0	57±0.1	**0±0**	**65.98±0.78**	0.5±1	65.51±2.05
	Densenet-121	Class-1	56.33±4.93	74.65±1.42	0±0	74.18±2.47	0±0	74.55±0.66	0±0	50.75±2.77	0±0	51.83±1.25	**0.5±0.58**	**63.96±1.37**
		Class-3	89.8±1.31	74.74±1.83	0±0	73.9±2.32	0±0	74.54±1.88	0±0	56.84±3.41	1.31±1.15	54.52±3.03	**0.15±0.17**	**66.38±3.65**
		Class-8	74.78±23.81	75.51±2.85	0±0	72.29±2.39	0±0	75.1±1.27	0±0	52.88±1.44	0.18±0.31	56.61±2.06	**0.4±0.46**	**64.6±3.51**
	ConvNeXt-L	Class-1	91.59±4.13	89.03±1.03	0±0	73.03±0.55	0±0	73.79±0.17	0±0	75.25±1.01	0±0	72.26±1.07	1.9±0.85	**76.82±1.19**
		Class-3	77.47±1.86	88.59±1.09	0±0	73.15±0.3	0±0	74.95±0.16	0±0	70.51±0.65	0±0	71±0.39	1±1.15	**72.51±1.12**
		Class-8	99.41±0.86	89.22±1.03	0±0	71.83±0.35	0±0	73.65±1.19	0±0	71.22±0.93	0±0	71.97±0.26	**0±0.18**	**72.64±0.23**
FOOD-101	Resnet-50	Class-10	67.2±5.54	78.18±0.01	0±0	75.1±0.23	0±0	77.3±0.77	0±0	60.83±1.4	0±0	56.07±1.08	**0.8±0.69**	**68.34±1.24**
		Class-30	90.8±4.16	77.94±0.01	0±0	74.86±0.09	0±0	77.08±0.25	0±0	62.13±0.95	0±0	52.45±0.57	**0.27±0.46**	**65.46±0.32**
		Class-50	59.6±4.85	78.26±0	0±0	74.18±0.2	0±0	77.23±0.32	0±0	63.06±0.77	0±0	55.53±0.48	**0±0**	**69.43±0.81**
	Densenet-121	Class-10	60.87±6.31	75.85±1.93	0±0	75.38±0.61	0±0	75.65±1.12	0±0	53.5±1.89	0±0	50.91±0.37	1.2±1.31	**64.97±2**
		Class-30	88.13±0.46	76.17±0.74	0±0	74.91±1.53	0±0	75.37±1.59	0±0	57.8±1.1	0.87±1.15	54.86±1.84	**0.67±0.86**	**64.75±2**
		Class-50	58.05±13.76	77.67±1.88	0±0	75.24±0.55	0±0	75.7±1.05	0±0	55.56±4.65	0.29±0.27	58.23±0.77	**0±0**	**67.9±2**
	ConvNeXt-L	Class-20	90.38±0.76	87.43±0.59	0±0	87.73±0.62	0±0	88.51±0.53	0±0	69.26±0.92	0±0	68.77±0.41	**0±0**	**75.97±0.79**
		Class-40	96.35±0.74	87.17±0.37	0±0	87.02±0.17	0±0	88.46±0.2	0±0	72.37±1.76	0±0	70.62±0.28	**0±0**	**78.45±0.79**
		Class-60	93.41±0.72	86.45±0.55	0±0	86.83±0.34	0±0	87.44±0.28	0±0	73.05±1.6	0±0	73.83±0.39	**0.74±0.5**	**81.79±0.99**

Table 1: accuracy on the forgotten class: $A_{D_f}(\%)$ and accuracy on the remaining classes: $A_{D_r}(\%)$ on MNIST, CIFAR-100 and FOOD-101 datasets for different models and unlearning-classes.

of information regarding a user's presence in the dataset. Ideally, the membership inference attack (MIA) accuracy should remain below 0.5, indicating that any unlearning mechanism should not render the model more susceptible to membership attacks than random chance.

- **Unlearning time:** The unlearning method is expected to be computationally inexpensive, incurring low computational cost. To assess computational efficiency, the method should yield satisfactory results within a limited number of epochs. Therefore, the runtime, measured in terms of epochs, serves as a crucial metric for evaluating various unlearning methods.

Unlearning Results

Experiments shown in Table 1 comprehensively detail the performance evaluation of different models: Resnet-34, Rensent-50, Densenet-121, and ConvNeXt-Large across three distinct datasets, considering various class unlearning settings. Across different datasets and models, our method consistently achieves lower accuracy on the forgotten class (A_{D_f}) during various unlearning tasks while the high accuracy on the remaining classes (A_{D_r}).

For the MNIST dataset, it is evident across various models and unlearning classes that all methods consistently attain minimal accuracy (approximately 0) on the unlearning class (A_{D_f}). Notably, our proposed method exhibits superior efficacy compared to other baseline methods, particularly in terms of retained class accuracy (A_{D_r}), achieving a level of performance akin to retraining with higher precision. In the context of the CIFAR-100 dataset, a similar pattern emerges. Nevertheless, our proposed method exhibits marginally elevated values for the unlearning class accuracy (A_{D_f}) in contrast to other baseline methods. This minor increase in A_{D_f} is accompanied by a noteworthy trade-off, manifesting as a substantial improvement of 1% to 20% in retained class accuracy (A_{D_r}) when compared to the baseline methods. This indicates that, while our method may have a slight compromise in unlearning effectiveness on the specified class, it compensates for this by significantly enhancing the preservation of knowledge pertaining to the retained classes. Similarly, for the FOOD-101 dataset, our proposed method demonstrates a comparable accuracy on the unlearning class (A_{D_f}) when juxtaposed with other baseline methods. However, a noteworthy distinction arises in the accuracy of the retrained classes (A_{D_r}), where our method excels by exhibiting an improvement of 5% to 10% over the baseline approaches. Additional results for other models and datasets can be found in Appendix Section B.

Any unlearning mechanism must avoid escalating the MIA accuracy beyond the level achievable by random chance. MIA accuracy variability depends upon the specific model, dataset, and unlearning class under consideration. The observations from Table 2 underscore that our proposed unlearning method consistently maintains MIA accuracy below 0.5. Resnet-18 exhibits a propensity for lower MIA accuracy when applied to the MNIST dataset. Conversely, Resnet-34 demonstrates relatively consistent performance across diverse datasets. Resnet-50, All-CNN, Densenet, and ConvNeXt consistently register MIA accuracy below 0.5, indicative of the effectiveness of our devised unlearning algorithm.

Further Fig. 2 shows the computational efficiency of our method compared to other baselines. Notably, our approach is a single-step method, eliminating the need for a repair step as seen in alternative approaches (Tarun et al. 2023).

Models	Datasets	—Unlearning Classes—	MIA accuracy
Resnet-18	MNIST	Class-2	0.24±0.01
	CIFAR-10	Class-3	0.49±0.01
	CIFAR-100	Class-1	0.48±0.03
	Food-101	Class-8	0.49±0
Resnet-34	MNIST	Class-8	0.42±0.02
	Cifar-10	Class-1	0.44±0
	Cifar-100	Class-3	0.44±0.01
	Food-101	Class-40	0.49±0
Resnet-50	Cifar-100	Class-8	0.49±0
	Food-101	Class-50	0.5±0
All-CNN	MNIST	Class-5	0.48±0.02
	Cifar-10	Class-6	0.48±0.01
Densenet	MNIST	Class-6	0.35±0.03
	CIFAR-10	Class-6	0.38±0.05
	CIFAR-100	Class-8	0.35±0.03
	Food-101	Class-30	0.39±0.04
ConvNeXt	MNIST	Class-2	0.39±0.04
	CIFAR-10	Class-2	0.37±0.07
	CIFAR-100	Class-3	0.39±0.04
	Food-101	Class-20	0.39±0.04

Table 2: Membership Inference Attack(MIA) accuracy of the unlearned models

This dual accomplishment of improved performance and decreased computational burden positions our method as a practical and efficient solution for class unlearning tasks.

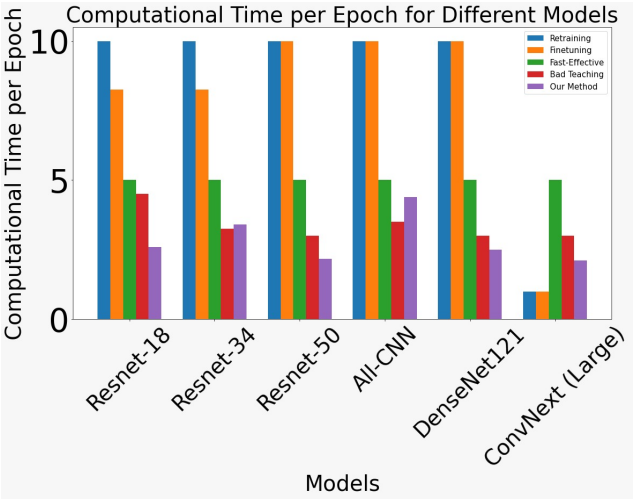

Figure 2: Unlearning Time comparison of our method with the Fast Yet Effective and Bad Teaching approach.

Ablation Study

Table 3 presents insights into the impact of a stability regularizer on different unlearning settings. In the absence of a stability regularizer, the accuracy on the unlearned class (A_{D_f}) is low (approaching 0), the accuracy of the model on the retained classes (A_{D_r}) experiences a significant decline. The findings emphasize that, without adequate stability regularization, models may struggle to maintain performance in the classes they were initially trained on.

Model	Dataset	Class	(α, β, γ)	A_{D_f}	A_{D_r}
Resnet-18	MNIST	Class-2	(631,0,0)	0±0	41.45±6.22
ConvNeXt	MNIST	Class-2	(1000,0,0)	0±0	14.56±3.71
All-CNN	CIFAR-10	Class-3	(2400,0,0)	7.4±3.08	60.59±3.71
Resnet-50	CIFAR-100	Class-3	(91,0,0)	0±0	15.83±15.35
Densenet-121	CIFAR-100	Class-3	(15000,0,0)	0±0	21.88±4.07
Resnet-50	Food-101	Class-30	(721,0,0)	0±0	15.09±10.19

Table 3: Effect of the stability regularizer on A_{D_f} and A_{D_r}: setting α to optimal and β, γ to zero

We conducted additional experiments, as presented in Table 4 where we held the values of β and γ constant at their optimal settings while systematically varying the value of α. Specifically, there is a consistent reduction in accuracy on the unlearned class (A_{D_f}) under the same number of training epochs. However, in contrast, the accuracy of the retained class (A_{D_r}) exhibits a declining trend.

Model	Dataset	Unlearn-Class	α	A_{D_f}	A_{D_r}
Resnet-18	**MNIST**	**Class-2**	100	5.85±0.48	95.95±0.16
			200	0.1±0.1	96.74±0.16
			400	0±0	97.24±0.3
			800	0±0	97.16±0.68
ConvNeXt	**MNIST**	**Class-2**	1000	0.17±0.29	98.48±0.59
			2000	0.14±0.24	98.63±0.22
			3000	0.07±0.12	98.05±0.51
			4000	0.17±0.3	97.43±0.9
All-CNN	**CIFAR-10**	**Class-3**	500	6.53±1.85	83.64±0.31
			1000	3.57±1.62	83.25±0.54
			2000	0±0	81.23±0.48
			4000	0.03±0.06	75.73±0.45
Resnet-50	**CIFAR-100**	**Class-3**	40	1.33±0.58	72.35±1.87
			80	0.33±0.58	72.72±0.57
			160	0±0	71.84±0.61
			320	0±0	71.52±0.24
Densenet-121	**CIFAR-100**	**Class-3**	40	1.6±2.77	69.15±4.1
			80	3.67±1.75	70.49±1.73
			120	2.33±1.62	68.96±2.2
			160	2±1.78	68.11±1.68
Resnet-50	**Food-101**	**Class-30**	40	1.87±1.15	70.27±0.64
			80	0.4±0	65.52±0
			160	0.8±1.39	69.64±0.48
			320	0.93±1.62	67.77±1.21

Table 4: Effect of α on A_{D_f} and A_{D_r}: Setting β, γ to optimal and varying α

Conclusion

With growing concerns related to privacy, safety, and model adaptability, our work presents a significant advancement in addressing fundamental challenges in machine unlearning. We present a novel method tailored for unlearning specific classes within deep classification models. A key distinguishing feature of our approach is its capability to function effectively even with partial access only to the unlearning class data. Unlike existing methods that may require multiple steps or complete dataset access for unlearning, our method achieves its objective in a single step. We believe that this unique characteristic significantly enhances the practicality and efficiency of the unlearning process, particularly in scenarios where full dataset access is restricted or impractical. Further, our method surpasses contemporaneous approaches consistently across a range of class unlearning challenges, underscoring its versatility and effectiveness.

Acknowledgments

Subhodip, a current Ph.D student at the ECE department of Indian Institute of Science (IISc), is supported by the Government of India via MOE fellowship. Subhodip also acknowledges the generous travel grant provided by the Kotak-IISc AI/ML Centre (KIAC) to attend The 39th Annual AAAI Conference on Artificial Intelligence. Shashwat, a current Physics Ph.D student at Washington University in St. Louis would like to acknowledge the financial support provided by the Data Science and Engineering department at his undergraduate institution Indian Institute of Science Education and Research (IISER), Bhopal to carry out this work as a BS thesis project at IISc. Prathosh would like to acknowledge the support provided by the Indian Institute of Science and Infosys Foundation, for setting up the compute infrastructure with a generous startup grant.

References

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, 446–461. Springer.

Brophy, J.; and Lowd, D. 2021. Machine Unlearning for Random Forests. *In Proc. of ICML*.

Cao, Y.; and Yang, J. 2015. Towards Making Systems Forget with Machine Unlearning. *In Proc. of IEEE Symposium on Security and Privacy*.

Challa, U. K.; Yellamraju, P.; and Bhatt, J. S. 2019. A multi-class deep all-CNN for detection of diabetic retinopathy using retinal fundus images. In *International Conference on Pattern Recognition and Machine Intelligence*, 191–199. Springer.

Chourasia, R.; and Shah, N. 2023. Forget Unlearning: Towards True Data-Deletion in Machine Learning. *In Proc. of ICML*.

Chundawat, V. S.; Tarun, A. K.; Mandal2, M.; and Kankanhalli, M. 2023. Can Bad Teaching Induce Forgetting? Unlearning in Deep Networks Using an Incompetent Teacher. *In Proc. of AAAI*.

Ginart, A.; Guan, M.; Valiant, G.; and Zou, J. Y. 2019. Making AI Forget You: Data Deletion in Machine Learning. *In Proc. of NIPS*.

Golatkar, A.; Achille, A.; Ravichandran, A.; Polito, M.; and Soatto, S. 2021a. Mixed-Privacy Forgetting in Deep Networks. *In Proc. of CVPR*.

Golatkar, A.; Achille, A.; Ravichandran, A.; Polito, M.; and Soatto, S. 2021b. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 792–801.

Golatkar, A.; Achille, A.; and Soatto, S. 2020a. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. *In Proc. of CVPR*.

Golatkar, A.; Achille, A.; and Soatto, S. 2020b. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9304–9312.

Golatkar, A.; Achille, A.; and Soatto, S. 2020c. Forgetting Outside the Box: Scrubbing Deep Networks of Information Accessible from Input-Output Observations. *In Proc. of ECCV*.

Golatkar, A.; Achille, A.; and Soatto, S. 2020d. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, 383–398. Springer.

Goldman, E. 2020. An introduction to the california consumer privacy act (ccpa). *Santa Clara Univ. Legal Studies Research Paper*.

Graves, L.; Nagisetty, V.; and Ganesh, V. 2021a. Amnesiac Machine Learning. *In Proc. of AAAI*.

Graves, L.; Nagisetty, V.; and Ganesh, V. 2021b. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11516–11524.

Guo, C.; Goldstein, T.; Hannun, A.; and van der Maaten, L. 2020. Certified data removal from machine learning models. *In Proc. of ICML*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsellal, R. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*.

Koh, P. W.; and Liang, P. 2017. Understanding Black-box Predictions via Influence Functions. *In Proc. of ICML*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Liu, B.; Liu, Q.; and Stone, P. 2022. CONTINUAL LEARNING AND PRIVATE UNLEARNING. *In Proc. of CoLLA*.

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A ConvNet for the 2020s. *arXiv e-prints*, arXiv:2201.03545.

Loo, N.; Swaroop, S.; and Turner, R. E. 2021. Generalized Variational Continual Learning. *arXiv*.

Ma, Z.; Liu, Y.; Liu, X.; Liu, J.; Ma, J.; and Ren, K. 2022. Learn to Forget: Machine Unlearning via Neuron Masking. *In Proc. of IEEE Transactions on Dependable and Secure Computing*.

Moulin, P.; and Veeravalli, V. V. 2019. *Statistical Inference for Engineers and Data Scientists*. Cambridge University Press.

Neel, S.; Roth, A.; and Sharifi-Malvajerdi, S. 2021. Descent-to-Delete: Gradient-Based Methods for Machine Unlearning. *In Proc. of ALT*.

Nguyen, C. V.; Li, Y.; Bui, T. D.; and Turner, R. E. 2017. Variational Continual Learning. *arXiv*.

Nguyen, Q. P.; Low, B. K. H.; and Jaillet, P. 2020. Variational Bayesian Unlearning. *In Proc. of NIPS*.

Nguyen, Q. P.; Oikawa, R.; Divakaran, D. M.; Chan, M. C.; and Low, B. K. H. 2022a. Markov Chain Monte Carlo-Based Machine Unlearning: Unlearning What Needs to be Forgotten. *In Proc. of ASIA CCS*.

Nguyen, T. T.; Huynh, T. T.; Nguyen, P. L.; Liew, A. W.-C.; Yin, H.; and Nguyen, Q. V. H. 2022b. A Survey of Machine Unlearning. *arXiv preprint arXiv:2209.02299*.

Pan, P.; Swaroop, S.; Immer, A.; Eschenhagen, R.; Turner, R.; and Khan, M. E. E. 2020. Continual Deep Learning by Functional Regularisation of Memorable Past. *Advances in Neural Information Processing Systems 33*.

Schwarz, J.; Czarnecki, W.; Luketina, J.; Grabska-Barwinska, A.; Teh, Y. W.; Pascanu, R.; and Hadsell, R. 2018. Progress and Compress: A scalable framework for continual learning. *Proceedings of the 35th International Conference on Machine Learning, PMLR 80:4528-4537*.

Sekhari, A.; Acharya, J.; Kamath, G.; and Suresh, A. T. 2021. Remember What You Want to Forget: Algorithms for Machine Unlearnings. *In Proc. of NeurIPS*.

Tanno, R.; Pradier, M. F.; Nori, A.; and Li, Y. 2022. Repairing Neural Networks by Leaving the Right Past Behind. *In Proc. of NeuRIPS*.

Tarun, A. K.; Chundawat, V. S.; Mandal, M.; and Kankanhalli, M. 2023. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*.

Voigt, P.; and dem Bussche, A. 2017. *The EU general data protection regulation (GDPR)*. Springer.

Wu, G.; Hashemi, M.; and Srinivasa, C. 2022. PUMA: Performance Unchanged Model Augmentation for Training Data Removal. *In Proc. of AAAI*.

Wu, Y.; Dobriban, E.; and Davidson, S. B. 2020. DeltaGrad: Rapid retraining of machine learning models. *In Proc. of ICML*.

Wu, Y.; Tannen, V.; and Davidson, S. B. 2020. PrIU: A Provenance-Based Approach for Incrementally Updating Regression Models. *In Proc. of SIGMOD*.

Xu, H.; Zhu, T.; Zhang, L.; Zhou, W.; and Yu, P. S. 2020. Machine Unlearning: A Survey. *ACM Computing Surveys Vol. 56, No. 1*.

Ye, J.; Fu, Y.; Song, J.; Yang, X.; Liu, S.; Jin, X.; Song, M.; and Wang, X. 2022. Learning with Recoverable Forgetting. *In Proc. of ECCV*.