Fast Machine Unlearning without Retraining through Selective Synaptic Dampening

Jack Foster[*1,2], **Stefan Schoepf**[*1], **Alexandra Brintrup**[1,2]

[1]University of Cambridge, Department of Engineering
[2]The Alan Turing Institute
{jwf40, ss2823, ab702}@cam.ac.uk

Abstract

Machine unlearning, the ability for a machine learning model to forget, is becoming increasingly important to comply with data privacy regulations, as well as to remove harmful, manipulated, or outdated information. The key challenge lies in forgetting specific information while protecting model performance on the remaining data. While current state-of-the-art methods perform well, they typically require some level of retraining over the retained data, in order to protect or restore model performance. This adds computational overhead and mandates that the training data remain available and accessible, which may not be feasible. In contrast, other methods employ a retrain-free paradigm, however, these approaches are prohibitively computationally expensive and do not perform on par with their retrain-based counterparts. We present Selective Synaptic Dampening (SSD), a novel two-step, post hoc, retrain-free approach to machine unlearning which is fast, performant, and does not require long-term storage of the training data. First, SSD uses the Fisher information matrix of the training and forgetting data to select parameters that are disproportionately important to the forget set. Second, SSD induces forgetting by dampening these parameters proportional to their relative importance to the forget set with respect to the wider training data. We evaluate our method against several existing unlearning methods in a range of experiments using ResNet18 and Vision Transformer. Results show that the performance of SSD is competitive with retrain-based post hoc methods, demonstrating the viability of retrain-free post hoc unlearning approaches.

Introduction

Modern machine learning (ML) models are trained on vast amounts of data, much of which may be sensitive, private, or copyrighted. To address threats posed by large-scale data collection, authorities are enacting data privacy regulations that afford individuals the right to request the deletion of their data (e.g., GDPR (Voigt and Von dem Bussche 2017)). Despite the increasing need to facilitate forgetting, there is much work to be done in designing such algorithms. The process of forgetting information within an ML model is referred to as machine unlearning.

The challenge of machine unlearning can be thought of as a multi-objective task, conducting forgetting without degrading model performance on the remaining data. Nguyen et al. (2022) refers to this trade-off in terms of design requirements, referring to performance preservation as keeping the model completeness, and unlearning efficiency as timeliness and light-weightiness. Timeliness is a key constraint, as full retraining of a model without the to-be-forgotten data would yield the desired results but doing so is time and resource-intensive. Similarly, light-weightiness refers to what preparation is necessary for the unlearning process, such as storing a list of samples and parameter updates for every training batch as in Graves, Nagisetty, and Ganesh (2021). This adds significant overhead and cannot be performed post-hoc.

Current state-of-the-art approaches rely on various retraining or fine-tuning steps in order to preserve model performance while unlearning the specified data (Tarun et al. 2023a,b; Chundawat et al. 2023a,b; Graves, Nagisetty, and Ganesh 2021). This can add overhead and, importantly, mandates that the training data be stored permanently.

In this paper, we propose Selective Synaptic Dampening (SSD), a retraining-free, post hoc unlearning approach to enable lightweight and timely unlearning. We achieve this by distinguishing between generalized and specialized information, prioritising the protection of generalized, broadly useful information while dampening parameters that are specialized towards to-be-forgotten samples. SSD builds on the finding that overparameterised ML models are prone to memorization of training data (Lee, Lee, and Shin 2011; Feldman 2020; Carlini et al. 2019). Thus, we contend that targeting this specialized information can induce forgetting while minimising influence on the generalization capability of the model. The remaining information in the model is generalized and therefore not violating individual privacy (e.g., the ability to detect the shape of a person versus detecting Jane Smith). We use the diagonal of the Fisher information matrix (FIM) to identify these specialized parameters. Golatkar, Achille, and Soatto (2020a) have also proposed a retraining-free unlearning approach based on the FIM, but as shown by Tarun et al. (2023b) and our own benchmarks, their approach does not meet key design criteria. First, the computational effort exceeds retraining and takes orders of magnitude longer than the unlearning method of Tarun et al.

[*]These authors contributed equally.

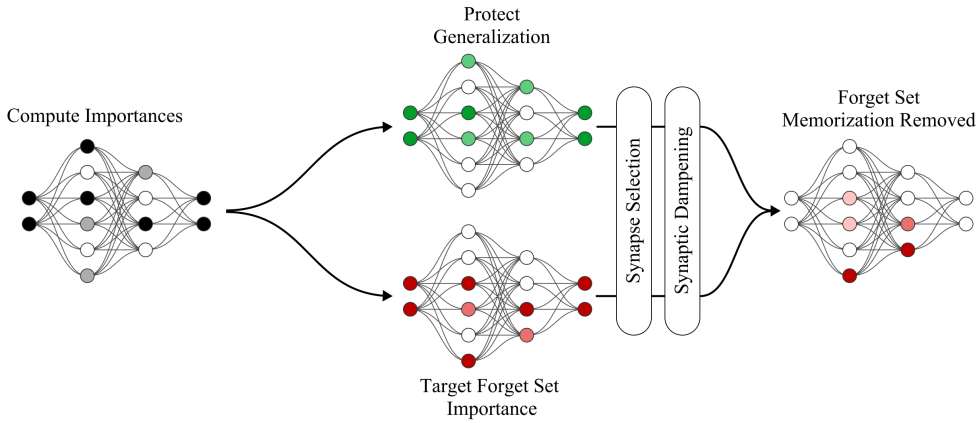

Figure 1: The Selective Synaptic Dampening process. Importance is calculated using the Fisher information. SSD identifies the parameters that are specialized towards the forget set, and dampens them proportional to this specialization

(2023b) based on their benchmarks. Second, their benchmarks also show that the unlearning performance does not match current state-of-the-art methods. We address both of these shortcomings via Selective Synaptic Dampening. Fig. 1 provides a diagrammatic overview of the SSD process.

We benchmark SSD against state-of-the-art unlearning methods (Chundawat et al. 2023a; Graves, Nagisetty, and Ganesh 2021; Tarun et al. 2023b) with three different unlearning scenarios: (i) single-class forgetting (Chundawat et al. 2023b), (ii) sub-class forgetting (Golatkar, Achille, and Soatto 2020b,c), (iii) random observations forgetting (Golatkar et al. 2021). Experimental results show that SSD is orders of magnitude faster than previous retrain-free methods (Golatkar, Achille, and Soatto 2020a,c) while performing comparably to established retrain-based methods both in terms of speed as well as forgetting performance.

We make the following key contributions:

1. We propose a novel retraining-free selective unlearning method that is competitive with state-of-the-art retraining-based methods.
2. We consider unlearning as a selective task in which only a small number of parameters should be modified to preserve model consistency.
3. SSD only needs access to the training data once to compute the FIM and can discard it afterwards, reducing storage requirements compared to retraining-based methods.

Related Work

Differential privacy. Differential privacy seeks to provide guarantees that information about individuals in a dataset is not leaked by the output of some model or function that uses this data (Dwork, Roth et al. 2014). Machine unlearning is strongly intertwined with this goal, with Ginart et al. (2019) introducing a probabilistic definition of unlearning that requires the output distribution of a model that has unlearnt data to be similar to the output distribution of a model that was never trained on that data.

Membership inference attacks (MIA). Deep learning models generally perform better on their training data than unseen data. Membership attacks exploit this to determine if a specific set of data was used in the training process by comparing model output distributions for test and train data Shokri et al. (2017); Hu et al. (2022). MIA is therefore a key measure of performance for unlearning methods.

Unlearning in deep networks. Due to the high cost of training for large models, and the need to be applied to existing models, we restrict our review to post hoc methods that do not require additional computations or data storage during the original training process (e.g., gradient vectors in Mehta et al. (2022), or a summation layer in (Cao and Yang 2015)). We categorise post hoc deep neural network unlearning methods into retraining-based and retraining-free approaches, based on whether they require any traditional model training steps in the unlearning procedure.

Retraining-free unlearning methods commonly utilise the Fisher information matrix. The FIM has long been used to approximate the sensitivity of a model's output to perturbations of its parameters from the second derivative of the loss (i.e. the Hessian), which can be interpreted as the importance of each parameter (as in Kirkpatrick et al. (2017) where it is used to calculate an L_2 regularization term to prevent forgetting of previous tasks). In unlearning, the FIM has been used in ad hoc (Guo et al. 2019), post hoc (Golatkar, Achille, and Soatto 2020a), and zero-shot (Sekhari et al. 2021) approaches. Golatkar, Achille, and Soatto (2020a) introduces Fisher Forgetting, a weight scrubbing method that induces forgetting by injecting noise into the parameters proportional to their relative importance to the forget set compared to the retain set. This is computationally very expensive and updates the whole model which causes significant degradation to the accuracy on the retain dataset, as shown in experiments of Tarun et al. (2023b). SSD addresses these shortcomings, yielding significantly faster execution time and through a stringent parameter-selection step, retain set performance is much better protected. Other non-FIM-based methods include variational forgetting for regression and Gaussian processes (Nguyen, Low, and Jaillet 2020), neural tangent kernel forgetting (NTK) (Golatkar, Achille, and Soatto 2020c), and mixed-linear models (MLM) (Go-

latkar et al. 2021). NTK and MLM rely on additional models that add further complexity and overhead. Selective Synaptic Dampening does not rely on any additional models.

Retraining-based unlearning methods are the current state-of-the-art in terms of performance. Chundawat et al. (2023a) uses a student-teacher framework with a competent and incompetent teacher model to induce forgetting while preserving model performance on the retained data. Graves, Nagisetty, and Ganesh (2021) present two unlearning methods of which one is post hoc which we will refer to as amnesiac in this paper. Amnesiac relabels $\mathcal{D}_f$ with randomly selected incorrect labels and then retrains the network for a set number of epochs. Tarun et al. (2023b) learns an error-maximising noise matrix for $\mathcal{D}_f$ that is then applied to the weights in the impair step before performing a repair step to recover model performance on $\mathcal{D}_r$. Chundawat et al. (2023b) and Tarun et al. (2023a) address the related yet distinct challenges of zero-shot and deep regression unlearning, respectively. We restrict the scope of this work to the more mature unlearning area of classification tasks.

Our method contrasts existing works by possessing a combination of desirable properties: post hoc, fast, retrain-free, selective in the parameters to be manipulated, and not reliant on additional models.

Preliminaries

Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ be a dataset of training samples x_i, with corresponding class label $y_i \in \{1, ..., K\}$. In an unlearning scenario, the objective is to forget the subset $\mathcal{D}_f \subset \mathcal{D}$, while preserving model performance on the remaining data $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$. We shall refer to these subsets as the forget set and retain set, respectively. $\mathcal{D}_f$ may comprise any subset of $\mathcal{D}$, and we show the performance of SSD on full class forgetting, where the forget set contains all samples with label k, subclass forgetting, where a subset of samples with label k are forgotten, and random subset, where each datapoint is randomly sampled from $\mathcal{D}$, without replacement. Let $\phi_\theta(\cdot) : X \to Y$, where $X \in \mathbb{R}^n$ and $Y \in \mathbb{R}^K$, be a function parameterised by $\theta \in \mathbb{R}^m$ and trained on $\mathcal{D}$, where the k-the component of $\phi_\theta(x)$ is the probability that sample x belongs to class k.

Proposed Method

The guiding intuition behind Selective Synaptic Dampening is that there likely exist parameters that are specifically important for $\mathcal{D}_f$ but not for $\mathcal{D}_r$. This intuition is further motivated by works such as Feldman (2020) and Stephenson et al. (2021). They show that deep neural networks memorize specific training examples and that parameters in later layers are highly specialized to specific features. Such parameters are likely extremely important for a small set of samples in the training data, but may not be generally important for the wider training set. Since $\mathcal{D}_r$ is typically large and filled with diverse samples, parameters which are similarly or more important for $\mathcal{D}_r$ compared to $\mathcal{D}_f$ likely correspond to highly generalized features, with little to no threat to differential privacy or the right to be forgotten. For example, recognising that there exists a person in an image is not necessarily a problem, but identifying *who* that person is, is a significant problem.

Hessian and the Fisher information matrix One way to identify important parameters is to use the FIM, as in Kirkpatrick et al. (2017); Golatkar, Achille, and Soatto (2020a); Guo et al. (2019); LeCun, Denker, and Solla (1989); Hassibi, Stork, and Wolff (1993). Given ϕ_θ, it can be assumed that the optimal parameters θ^* have been learnt, which minimises the loss over $\mathcal{D}$. The sensitivity of ϕ_θ with respect to each parameter θ_k can be calculated via the second-order derivative of the loss near the minimum (Maltoni and Lomonaco 2019). This sensitivity can be interpreted as the importance of each parameter. Calculating the second derivative is expensive, however, the diagonal of the Fisher information matrix is equivalent to the second derivative of the loss (Pawitan 2001), and critically can be computed using first-order derivatives. The FIM, and its first-order derivative property (Kay 1993; Aich 2021), is given in Eq. 1.

$$[]_{\mathcal{D}} = \mathbb{E}\left[-\frac{\delta^2 \, ln \, p(\mathcal{D}|\theta)}{\delta\theta^2} \mid_{\theta^*_{\mathcal{D}}}\right]$$

$$[]_{\mathcal{D}} = \mathbb{E}\left[\left(\left(\frac{\delta \, ln \, p(\mathcal{D}|\theta)}{\delta\theta}\right)\left(\frac{\delta \, ln \, p(\mathcal{D}|\theta)}{\delta\theta}\right)^T\right) \mid_{\theta^*_{\mathcal{D}}}\right] \quad (1)$$

Selective Synaptic Dampening We begin by outlining a naïve forgetting approach using the FIM in Eq. 2

$$\theta_i = \begin{cases} 0, & \text{if } []_{\mathcal{D}_{f,i}} > 0 \\ \theta_i, & \text{if } []_{\mathcal{D}_{f,i}} = 0 \end{cases} \quad \forall i \in [0, |\theta|] \quad (2)$$

where $[]_{\mathcal{D}_f,i}$ is the i-th element of the diagonal of the Fisher information matrix, calculated over the forget set $\mathcal{D}_f$. This represents a simple pruning algorithm, which identifies the location of all parameters that have non-zero importance values, and sets their value to zero, thereby removing their contribution to the model output. While this would lead to forgetting over $\mathcal{D}_f$, it would also lead to the catastrophic degradation of performance on $\mathcal{D}_r$, due to the large overlap in important parameters for both sets and the fact it is highly likely that $[]_{\mathcal{D}_f}$ is greater than zero for a majority of parameters. The challenge, then, lies in maintaining the forgetting abilities of such a pruning algorithm while simultaneously protecting parameters important to the retain set. To achieve this, we introduce two significant amendments to the pruning algorithm that lead to strong forgetting and retain-set performance while maintaining fast execution time. First, a stricter selection criterion is implemented, considering the parameter importance to the retain set in Eq. 3

$$\theta_i = \begin{cases} 0, & \text{if } []_{\mathcal{D}_{f,i}} > \alpha[]_{\mathcal{D}_r,i} \\ \theta_i, & \text{if } []_{\mathcal{D}_{f,i}} \leq \alpha[]_{\mathcal{D}_r,i} \end{cases} \quad \forall i \in [0, |\theta|] \quad (3)$$

where, the hyper-parameter α allows control of how protective the selection should be. The updated selection criteria now greatly reduces the number of parameters chosen, only selecting parameters that are more important for $\mathcal{D}_f$ than $\mathcal{D}_r$. This step facilitates the identification of parameters that are highly specialized towards samples in the forget set, with α dictating how specialized they must be to be pruned.

While this step is critical, there remains a clear limitation with this approach, which is the binary nature of the update rule. A parameter that is slightly over the threshold is treated the same as a parameter that is vastly more important to the forget set. This lack of granularity limits the forgetting-performance trade-off and necessitates a large α to maintain performance on $\mathcal{D}_r$, however a large α then significantly reduces the ability to forget $\mathcal{D}_f$ due to an unreasonably high bar for being considered specialized. Therefore, the pruning step is replaced by a dampening step that applies a penalty to the magnitude of the parameter proportional to its relative importance of $\mathcal{D}_f$ compared to $\mathcal{D}$ in Eq. 4

$$\beta = min(\frac{\lambda[]_{\mathcal{D},i}}{[]_{\mathcal{D}_{f,i}}}, 1)$$

$$\theta_i = \begin{cases} \beta\theta_i, & \text{if } []_{\mathcal{D}_{f,i}} > \alpha[]_{\mathcal{D},i} \\ \theta_i, & \text{if } []_{\mathcal{D}_{f,i}} \leq \alpha[]_{\mathcal{D},i} \end{cases} \quad \forall i \in [0, |\theta|] \tag{4}$$

where λ is a hyper-parameter to control the level of protection. This is the final SSD procedure. Intuitively, if $\lambda = 1$ then $\beta < 1$ for all parameters that are specialized towards $\mathcal{D}_f$. Therefore, $\beta \to 0$ as a parameter becomes more specialized for $\mathcal{D}_f$. Since λ scales this update, this dampening factor is given an upper bound of 1 to prevent large λ values from causing parameters to grow. The dampening effect, combined with the selection criteria, creates a granular method to forgetting that will almost completely remove highly-specialized parameters, protect generalized parameters, and proportionally dampen the parameters in between, thereby finding an acceptable compromise to this multi-objective problem. λ and α offer control over whether to prioritise forgetting or protecting, as well as performance adjustments for different models and unlearning tasks. Finally, we highlight that $[]_{\mathcal{D}_r}$ is substituted with $[]_{\mathcal{D}}$ in the selection step. This is because $[]_{\mathcal{D}_r}$ must be recalculated for every new forget request. $[]_{\mathcal{D}}$ can be calculated at any point after training before unlearning and only needs to be computed once, allowing for the training set to be discarded and only $[]_{\mathcal{D}}$ stored. With this substitution, the selection criteria can be thought of as trying to prevent the dampening from moving the parameter set away from the original, optimal parameters $\theta^*_{\mathcal{D}}$. This is a trade-off to optimise for speed of execution, and the solution remains accurate as typically $|\mathcal{D}_f| << |\mathcal{D}|$ and therefore the values for $[]_{\mathcal{D}}$ and $[]_{\mathcal{D}_r}$ are near identical. We hypothesise that this is a valid approach for repeated forget requests, as only a small fraction of parameters are updated for each forget request, and therefore it would take many forget requests for $[]_{\mathcal{D}}$ and $[]_{\mathcal{D}_r}$ to diverge substantially. In the event of such divergence, the only consequence would be the false protection of now-purged parameters, since dampening will only reduce a parameter's contribution to model output.

The experimental results are all calculated with $\mathcal{D}$ rather than $\mathcal{D}_r$, and demonstrate the efficacy of this approach.

Algorithm 1: Selective Synaptic Dampening

Input: ϕ_θ, $\mathcal{D}$, $\mathcal{D}_f$; optional to skip 1.: $[]_{\mathcal{D}}$
Parameter: α, λ
Output: $\phi_{\theta'}$

```
Calculate and store []_D once. Discard D.
Calculate []_{D_f}
for i in range |θ| do
    if []_{D_f,i} > α[]_{D,i} then
        θ'_i = min(λ[]_{D,i} / []_{D_f,i} θ_i, θ_i)
    end if
end for
return φ_{θ'}
```

Experimental Setup

Datasets used. We evaluate our method on image classification using CIFAR10, CIFAR20, and CIFAR100 (Krizhevsky and Hinton 2010), in line with Golatkar, Achille, and Soatto (2020a); Chundawat et al. (2023a). We forget the same classes from these datasets as (Chundawat et al. 2023a). Golatkar, Achille, and Soatto (2020a) also use the VGG-Face dataset (Parkhi, Vedaldi, and Zisserman 2015), however, this is no longer accessible so we substitute it with the PinsFaceRecognition dataset (Burak 2020), which consists of 17,534 faces of 105 celebrities collected from Pinterest.

Models used. Following Chundawat et al. (2023a), we use ResNet18 (He et al. 2016) and Vision Transformer (Dosovitskiy et al. 2021) for the learning and unlearning tasks. Experiments were performed on NVIDIA RTX4090 with Intel Xeon processors. Models are trained with early stopping using a multi-step learning rate scheduler beginning at $lr = 0.1$ and the Adam optimiser (Kingma and Ba 2014) with Python 3, PyTorch, and Ubuntu 20.04.6 LTS.

Evaluation Measures. Analogous to Chundawat et al. (2023a), we use the following: 1) *Accuracy on the forget and retain set:* To validate forgetting while retaining overall model performance. Listed as $\mathcal{D}_f$ and $\mathcal{D}_r$ in results tables. 2) *Membership inference attack:* To investigate if information about the forget sample is still present in the model. We use the logistic regression MIA implementation from Chundawat et al. (2023a). We also consider an additional metric 3) *Execution time (seconds):* to evaluate the timeliness of methods (denoted t in results).

Unlearning tasks: We benchmark across three different unlearning scenarios: (i) Single-class forgetting (Chundawat et al. 2023b), (ii) sub-class forgetting (Golatkar, Achille, and Soatto 2020b,c), and (iii) random observations forgetting (Golatkar et al. 2021). In (i) we forget a superclass of CIFAR 20, as well as a class out of CIFAR100 and PinsFaceRecognition. In (ii) we forget a CIFAR 20 subclass of a superclass (e.g., rocket out of vehicles). CIFAR20 superclasses have CIFAR100 classes as subclasses. In (iii) we forget a random subset of 100 samples from CIFAR10.

Baselines used. We compare SSD to the Fisher Forgetting algorithm (Golatkar, Achille, and Soatto 2020a), however initial results show that not only does Fisher Forgetting perform worse than SSD, it also is exceptionally slow (between $50 - 250$ times slower than SSD). Similarly, Nguyen, Low, and Jaillet (2020); Golatkar, Achille, and Soatto (2020c); Golatkar et al. (2021) fit additional compute-intensive mod-

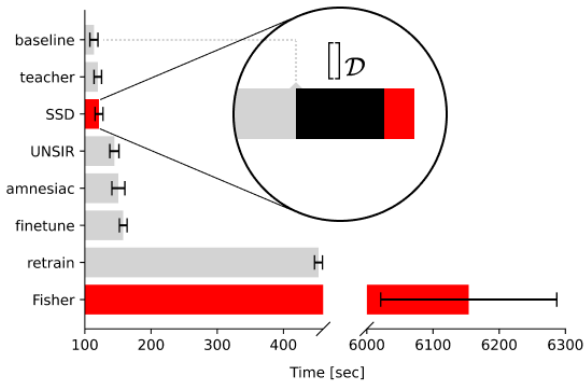

Figure 2: Time per method on (ii) Cifar20 rocket including model loading and metric calculation (i.e. baseline) to simulate a realistic forget process from model loading to verification of forgetting. Zoom-in indicates share of compute time spent on calculating $[]_{\mathcal{D}}$ compared to the rest of the SSD method. Precomputing $[]_{\mathcal{D}}$ can save 84.32%±0.21% of the pure SSD computing time (excl. baseline time).

els or perform computations exceeding Fisher in computational time (e.g., NTK), thus rendering them prohibitively expensive to run over all experiments. Therefore, we focus our comparison on state-of-the-art methods, which are in the retraining-based category. We compare SSD against the following methods: (a) *Baseline:* The unaltered model trained on $\mathcal{D}_r \cup \mathcal{D}_f$ (b) *Finetune:* Finetuning the baseline model on $\mathcal{D}_r$ for 5 epochs, (c) *Retraining:* Retraining the model from scratch on $\mathcal{D}_r$, (d) *Bad Teacher (Chundawat et al. 2023a)*, (e) *Amnesiac (Graves, Nagisetty, and Ganesh 2021)*, (f) *UNSIR (Tarun et al. 2023b)*. (f) is not designed for random observations forgetting and therefore excluded from this task.

SSD parameters. We found hyper-parameters using 50 runs of the TPE search from Optuna (Akiba et al. 2019), for values $\alpha \in [0.1, 100]$) and $\lambda \in [0.1, 5]$. We only conducted this search for the Rocket and Veh2 classes. We use λ=1 and α=10 for all ResNet18 CIFAR tasks. For PinsFaceRecognition, we use α=50 and λ=0.1 due to the much greater similarity between classes. ViT also uses λ=1 on all CIFAR tasks. We change α=10 to α=5 for slightly improved performance on class and α=25 on sub-class unlearning.

Results and Discussion

Defining good. Chundawat et al. (2023a) note that theoretically perfect accuracy $\mathcal{D}_f = 0$ and $MIA = 0$ is not necessarily ideal. Ideal is to closely match the performance of a model retrained from scratch that has never seen $\mathcal{D}_f$, the *gold model*. They postulate that deviating from the gold model performance can lead to the *Streisand effect*. They give the example of a model that classifies a Boeing aircraft as a mushroom. This is maximally wrong but not a behaviour expected from a model that can classify other aircraft. The model thus leaks information to an attacker by deliberately being wrong. The model has not unlearned, it simply learned to predict a wrong label for $\mathcal{D}_f$. This is congruent with the probabilistic definition of unlearning from Ginart et al. (2019). This is especially relevant in the (iii) random forgetting scenario, where the distributions of $\mathcal{D}_r$ and $\mathcal{D}_f$ are likely to be very similar, and an unlearned sample from the rocket class could still be correctly classified based on the generalized knowledge about rockets in the model from the $\mathcal{D}_r$ rocket samples. Therefore, we define good as unlearning matching the MIA of the retrained model.

		retrain	Fisher	SSD
Cifar20 Veh2	$\mathcal{D}_r$	**82.11±0.19**	5.76±1.01	**82.97±0.00**
	$\mathcal{D}_f$	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
	MIA	**13.54±0.01**	47.12±16.39	**6.68±0.00**
	t	441±10	5871±297	**122±5**
Cifar100 Rocket	$\mathcal{D}_r$	**72.83±0.42**	1.18±0.06	**74.54±0.00**
	$\mathcal{D}_f$	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
	MIA	**1.04±0.00**	**0.00±0.16**	2.20±0.00
	t	1805±10	28744±1332	**120±6**
Cifar20 Rocket	$\mathcal{D}_r$	**81.54±0.24**	5.20±0.54	**82.43±0.00**
	$\mathcal{D}_f$	**10.74±3.40**	0.78±1.35	**2.17±0.00**
	MIA	**3.85±0.01**	43.40±7.79	**10.80±0.00**
	t	453±6	6154±133	**121±6**
Cifar10 Random	$\mathcal{D}_r$	**91.45±0.11**	12.74±2.22	**88.68±3.36**
	$\mathcal{D}_f$	**94.10±2.00**	11.85±4.13	**93.61±4.99**
	MIA	**74.22±0.04**	46.59±27.46	**72.65±0.05**
	t	308±5	3225±100	**121±5**

Table 1: Fisher unlearning on (i), (ii), and (iii) tasks with ResNet18. As reported by Tarun et al. (2023b), we also observe that Fisher fails to maintain accuracy on the retained data and is computationally very expensive. SSD is deterministic, thus no ± is reported. Veh2: Vehicle2. $\mathcal{D}_r$ and $\mathcal{D}_f$ rows report the accuracy on the respective dataset. All values in percent [%] except t [seconds]. For retrain, $D_f > 0$ due to model generalization (e.g., rocket similar to vehicles)

Selectivity. SSD only changes a small amount of parameters. When forgetting the rocket class from Cifar100, only 1.7% of parameters are changed.

Comparison to Fisher Forgetting. Table 1 shows a comparison of Fisher (Golatkar, Achille, and Soatto 2020a), a retrained model, and SSD. Experimental results show that SSD significantly outperforms Fisher in terms of closeness to the retrained model performance, with Fisher significantly dropping $\mathcal{D}_r$ performance, and performing considerably worse on the MIA evaluation for the Cifar20 single-class and Cifar20 sublcass tasks. Furthermore, SSD is orders of magnitude faster than Fisher, only requiring $0.4 - 3.8$ percent of the time that Fisher requires. We also note that Fisher's execution time increases significantly with larger models, whereas SSD is less sensitive to this as shown with Cifar100 rocket class unlearning times of 120.00±5.49 seconds on ResNet8 (11M parameters) and 655.64±65.36 seconds on ViT (85M parameters).

Compute time comparison. Fig. 2 shows compute times for the Cifar20 class unlearning task. We experimentally demonstrate that SSD is comparable to state-of-the-art meth-

	Class	metric	baseline	retrain	finetune	teacher	UNSIR	amnesiac	SSD
RN	rocket	$\mathcal{D}_r$	**76.27±0.00**	72.83±0.42	64.05±0.88	74.53±0.26	73.89±0.28	73.34±0.45	**74.54±0.00**
		$\mathcal{D}_f$	80.90±0.00	**0.00±0.00**	0.00±0.00	**0.00±0.00**	28.66±4.98	**0.00±0.00**	**0.00±0.00**
		MIA	93.40±0.00	**1.04±0.41**	13.70±0.04	**0.00±0.00**	1.94±0.01	29.56±0.02	2.20±0.00
	MR	$\mathcal{D}_r$	**76.28±0.00**	72.90±0.45	63.97±0.67	74.53±0.26	73.81±0.26	73.56±0.48	**75.59±0.00**
		$\mathcal{D}_f$	80.12±0.00	**0.00±0.00**	0.00±0.00	**0.00±0.00**	27.34±5.08	**0.00±0.00**	**0.00±0.00**
		MIA	95.20±0.00	**0.22±0.01**	12.98±0.03	0.00±0.00	1.54±0.01	46.48±0.04	**0.20±0.00**
ViT	rocket	$\mathcal{D}_r$	**88.88±0.00**	90.07±0.09	80.82±1.37	87.46±0.53	88.47±0.38	87.92±0.89	**88.90±0.00**
		$\mathcal{D}_f$	94.70±0.00	**0.00±0.00**	0.46±0.72	4.20±5.24	65.32±9.11	**0.00±0.00**	**0.00±0.00**
		MIA	94.40±0.00	**3.23±0.50**	19.00±0.09	0.03±0.00	29.13±0.06	1.00±0.01	**1.80±0.00**
	MR	$\mathcal{D}_r$	88.87±0.00	**90.02±0.22**	81.14±0.79	87.42±0.41	88.44±0.58	88.34±0.72	**88.82±0.00**
		$\mathcal{D}_f$	94.88±0.00	**0.00±0.00**	2.33±2.37	12.82±5.92	83.94±2.87	**0.00±0.00**	**0.00±0.00**
		MIA	92.80±0.00	**0.70±0.41**	7.10±0.02	0.03±0.00	21.33±0.03	**0.47±0.00**	3.80±0.00

Table 2: (i) Class unlearning on CIFAR100 with ResNet18 (RN) and Vision Transformer (ViT). MR: mushroom. $\mathcal{D}_r$ and $\mathcal{D}_f$ rows report the accuracy on the respective dataset. All values in percent [%].

	Class	metric	baseline	retrain	finetune	teacher	UNSIR	amnesiac	SSD
RN	Veh2	$\mathcal{D}_r$	**82.69±0.00**	82.11±0.19	73.50±0.86	81.96±0.21	80.81±0.46	81.13±0.3	**82.97±0.00**
		$\mathcal{D}_f$	80.41±0.00	**0.00±0.00**	0.00±0.00	3.62±1.07	46.92±2.27	**0.00±0.00**	**0.00±0.00**
		MIA	82.56±0.00	**13.54±0.01**	30.63±0.04	0.00±0.00	35.16±0.03	**7.54±0.01**	6.68±0.00
	veg	$\mathcal{D}_r$	**82.31±0.00**	81.39±0.21	71.42±1.32	81.46±0.3	80.29±0.26	81.01±0.33	**82.38±0.00**
		$\mathcal{D}_f$	86.90±0.00	**0.00±0.00**	0.00±0.00	2.67±1.35	64.45±1.77	**0.00±0.00**	**0.00±0.00**
		MIA	89.52±0.00	**9.74±0.01**	29.39±0.08	0.00±0.00	40.66±0.06	**5.00±0.01**	16.96±0.00
ViT	Veh2	$\mathcal{D}_r$	**95.73±0.00**	94.85±0.13	87.75±1.64	93.59±0.3	93.56±0.32	**93.88±0.15**	93.12±0.00
		$\mathcal{D}_f$	95.22±0.00	**0.00±0.00**	0.04±0.12	4.88±4.12	70.31±5.03	**0.00±0.00**	**0.00±0.00**
		MIA	84.04±0.00	**22.96±0.03**	38.15±0.08	0.02±0.00	48.98±0.07	1.20±0.00	**7.04±0.00**
	veg	$\mathcal{D}_r$	**95.59±0.00**	94.54±0.21	87.09±1.24	92.92±0.51	93.25±0.35	93.29±0.41	**95.71±0.00**
		$\mathcal{D}_f$	97.57±0.00	**0.00±0.00**	0.30±0.29	8.28±6.79	89.02±2.41	0.02±0.07	**0.00±0.00**
		MIA	91.32±0.00	**4.41±0.01**	14.72±0.05	0.02±0.00	58.67±0.04	1.02±0.00	**1.88±0.00**

Table 3: (i) Class unlearning on CIFAR20 with ResNet18 and Vision Transformer. Veh2: Vehicle2.

	metric	baseline	retrain	finetune	teacher	UNSIR	amnesiac	SSD
RN	$\mathcal{D}_r$	98.52±0.02	**100.00±0.00**	99.72±0.45	96.72±0.44	99.89±0.06	**99.99±0.02**	98.42±0.13
	$\mathcal{D}_f$	97.84±1.99	**0.00±0.00**	4.32±4.61	0.13±0.4	90.53±5.68	**0.00±0.00**	**0.00±0.00**
	MIA	34.38±0.23	**0.00±0.00**	0.80±0.01	**0.02±0.00**	8.54±0.11	8.92±0.03	1.11±0.01

Table 4: (i) Face unlearning. One face unlearned per experiment [ID 1,10,20,30,40] and results aggregated for 5 experiments.

ods. For repeated unlearning, as expected in practice, $[]_{\mathcal{D}}$ can be computed once and stored, reducing the time far below the already competitive time, which would make SSD the fastest method. Including the computation of $[]_{\mathcal{D}}$, SSD is the second fastest method behind Chundawat et al. (2023a).

(i) Class unlearning. SSD is first benchmarked on class unlearning, as performed in Chundawat et al. (2023a), on CIFAR100 in Table 2, CIFAR20 in Table 3, and PinsFaceRecognition unlearning in Table 4. SSD is close to the retrained model in terms of $\mathcal{D}_r$ and MIA across the unlearning tasks and is comparable to retraining-based methods. For example, forgetting rocket from Cifar100 has a baseline MIA of 93% for ResNet and 94% for ViT, SSD reduces this to ca. 2% (retrain 1-3$\%$), while $\mathcal{D}_r$ performance drops just 2% for ResNet, and actually improves negligibly for ViT. We highlight the closest method to retrain bold in the results tables.

(ii) Subclass unlearning. We present subclass unlearning, as performed in Chundawat et al. (2023a), on CIFAR20 in Table 5. Class *sea* demonstrates the problem of defining good, as the retrained model achieves a high non-zero MIA. Amnesiac and Bad Teacher reduce MIA to near zero, even though a retrained model does not show the same behaviour. Graves, Nagisetty, and Ganesh (2021) relabel $\mathcal{D}_f$ to random labels and (Chundawat et al. 2023a) uses an incompetent teacher to update the model. The noise-based approach of Tarun et al. (2023b) and our SSD on the other hand lead to higher MIA and $\mathcal{D}_f$ values that are closer to the retrained model. Efficacy analysis is therefore hard, as while Amnesiac and Teacher minimise the MIA and $\mathcal{D}_f$ accuracy, they may be falling victim to the *Streisand effect*.

	Class	metric	baseline	retrain	finetune	teacher	UNSIR	amnesiac	SSD
RN	rocket	$\mathcal{D}_r$	**82.54±0.00**	81.54±0.24	72.41±0.95	81.48±0.27	81.13±0.31	81.46±0.26	**82.43±0.00**
		$\mathcal{D}_f$	79.34±0.00	**10.74±3.4**	9.75±6.68	**6.41±3.57**	59.20±4.75	0.76±0.73	2.17±0.00
		MIA	89.40±0.00	**3.85±0.01**	18.67±0.05	0.00±0.00	33.53±0.06	**6.60±0.01**	10.80±0.00
	sea	$\mathcal{D}_r$	**82.37±0.00**	81.30±0.27	72.50±1.55	81.22±0.24	80.82±0.3	81.05±0.31	**81.72±0.00**
		$\mathcal{D}_f$	96.27±0.00	**91.47±1.92**	82.69±7.17	75.13±4.12	**95.49±2.4**	46.78±8.55	75.35±0.00
		MIA	90.80±0.00	**52.09±0.03**	62.82±0.11	0.00±0.00	**80.44±0.04**	4.45±0.01	21.80±0.00
ViT	rocket	$\mathcal{D}_r$	**95.73±0.00**	94.61±0.13	85.70±3.05	93.60±0.29	93.34±0.45	93.47±0.22	**95.13±0.00**
		$\mathcal{D}_f$	94.53±0.00	**22.26±8.34**	6.25±6.03	3.35±2.89	74.93±10.13	0.85±1.71	**5.12±0.00**
		MIA	80.40±0.00	**3.44±0.01**	16.04±0.03	0.02±0.00	27.27±0.14	**0.78±0.00**	5.40±0.00
	sea	$\mathcal{D}_r$	**95.67±0.00**	94.55±0.22	87.65±1.56	93.57±0.26	93.26±0.31	93.26±0.24	**95.57±0.00**
		$\mathcal{D}_f$	99.22±0.00	**95.12±0.81**	89.17±4.17	25.97±14.01	**94.25±2.32**	21.42±8.5	97.05±0.00
		MIA	88.40±0.00	**65.96±0.04**	65.04±0.13	0.17±0.00	**76.96±0.07**	0.40±0.00	82.20±0.00

Table 5: (ii) Subclass unlearning on CIFAR20 with ResNet18 and Vision Transformer.

	metric	baseline	retrain	finetune	teacher	amnesiac	SSD
RN	$\mathcal{D}_r$	90.71±0.00	**91.45±0.11**	88.02±0.45	**90.21±0.10**	90.16±0.23	88.68±3.36
	$\mathcal{D}_f$	95.30±2.08	**94.10±2.00**	90.00±3.73	90.00±2.73	59.04±4.79	**93.61±4.99**
	MIA	75.78±0.04	**74.22±0.04**	74.58±0.05	49.28±0.07	25.18±0.05	**72.65±0.05**
ViT	$\mathcal{D}_r$	**98.88±0.00**	98.61±0.08	97.28±0.33	97.58±0.36	97.62±0.35	**98.01±1.56**
	$\mathcal{D}_f$	100.00±0.00	**98.80±0.76**	97.19±0.98	86.75±3.57	73.49±5.11	**98.07±2.35**
	MIA	90.76±0.03	**91.77±0.02**	86.14±0.02	33.53±0.06	10.44±0.05	**85.54±0.11**

Table 6: (iii) Random unlearning on CIFAR10 with ResNet18 and Vision Transformer.

(iii) Random sample unlearning. We present random sample unlearning, as performed in Golatkar, Achille, and Soatto (2020a), on CIFAR10 in Table 5. We observe similar performance of SSD and retraining on ResNet and ViT. Graves, Nagisetty, and Ganesh (2021) and Chundawat et al. (2023a) again reduce the MIA far below the retrained model as observed in (ii), which is a risk to privacy (*Streisand effect*). An ideal unlearned model mimics a retrained model.

Overall analysis of Selective Synaptic Dampening. SSD outperformed Fisher Forgetting while being orders of magnitude faster, highlighting its efficacy. SSD also performs competitively with established state-of-the-art methods and full model retraining, demonstrating the viability of retrain-free post-hoc unlearning approaches in a wider context. SSD was, on average, the strongest performing method when measuring similarity to the fully retrained model. However, the lack of standardized evaluations in unlearning, and an as-yet-undecided notion of what is truly a *good* MIA score, renders a qualitative assessment of methods challenging and determining the best algorithm ambiguous.

Limitations. First, SSD is not certified, with no mathematical guarantee of unlearning a given sample. This weakness is shared by all benchmarked methods. Second, if bad parameter values are chosen (α, λ), such that large changes are made to the model, then repeat forgetting may lead to significant model degradation. Finding appropriate values for α and λ is a practical limitation but as shown experimentally, the parameters are only set within one order of magnitude ($\alpha \in [5, 50]$ and $\lambda \in [0.1, 1]$) across two models of vastly different parameter counts and architectures. We hypothesise that the ideal parameters could be estimated from the $\mathcal{D}_f$ loss distribution to enable automatic parameter selection in future work. Finally, we note that without a repair step, there is naturally a finite amount of forget requests that SSD can process before $\mathcal{D}_r$ performance begins to degrade.

Conclusion

We present a novel two-step, retraining-free unlearning method. SSD first selects parameters that are considerably more important for the forget set than the retain set, before dampening these parameters proportional to the discrepancy in their importance to the forget and retain set. The result of these steps is a fast yet highly effective method for machine unlearning. We evaluate SSD on a range of tasks, demonstrating viability in single-class, sub-class and random sample settings, on multiple datasets and different model architectures. Results show that SSD is orders of magnitude faster than the comparable Fisher Forgetting method, outperforming the method considerably; SSD even rivals the speed and performance of state-of-the-art retrain-based approaches.

Many future directions could be explored, such as evaluating how to improve and measure performance on random subsets, given the significant overlap in parameter importance for the forget and test set. Another interesting direction is how to forget large subsets of information. Typically experiments evaluate forgetting no more than 5-10% of data; this may be realistic but evaluating how to increase the upper bound of forgetting without retraining may offer valuable insight into how to improve existing unlearning methods.

Acknowledgments

This work was supported by the Accenture Turing Strategic Partnership, the Turing Enrichment scheme, EPSRC CDT AgriFoRwArdS [grant number EP/S023917/1], and EPSRC DTP [grant number EP/W524633/1].

References

Aich, A. 2021. Elastic weight consolidation (EWC): Nuts and bolts. *arXiv preprint arXiv:2105.04093*.

Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; and Koyama, M. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.

Burak. 2020. Pinterest Face Recognition Dataset. kaggle.com/datasets/hereisburak/pins-face-recognition. Accessed: 2023-08-09.

Cao, Y.; and Yang, J. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, 463–480. IEEE.

Carlini, N.; Liu, C.; Erlingsson, Ú.; Kos, J.; and Song, D. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, 267–284.

Chundawat, V. S.; Tarun, A. K.; Mandal, M.; and Kankanhalli, M. 2023a. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 7210–7217.

Chundawat, V. S.; Tarun, A. K.; Mandal, M.; and Kankanhalli, M. 2023b. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4): 211–407.

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

Ginart, A.; Guan, M.; Valiant, G.; and Zou, J. Y. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32.

Golatkar, A.; Achille, A.; Ravichandran, A.; Polito, M.; and Soatto, S. 2021. 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9304–9312.

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 (CVPR)*.

Golatkar, A.; Achille, A.; and Soatto, S. 2020c. 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.

Graves, L.; Nagisetty, V.; and Ganesh, V. 2021. 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. 2019. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*.

Hassibi, B.; Stork, D. G.; and Wolff, G. J. 1993. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, 293–299. IEEE.

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.

Hu, H.; Salcic, Z.; Sun, L.; Dobbie, G.; Yu, P. S.; and Zhang, X. 2022. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s): 1–37.

Kay, S. M. 1993. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Krizhevsky, A.; and Hinton, G. 2010. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7): 1–9.

LeCun, Y.; Denker, J.; and Solla, S. 1989. Optimal brain damage. *Advances in neural information processing systems*, 2.

Lee, J.; Lee, J.-N.; and Shin, H. 2011. The long tail or the short tail: The category-specific impact of eWOM on sales distribution. *Decision Support Systems*, 51(3): 466–479.

Maltoni, D.; and Lomonaco, V. 2019. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116: 56–73.

Mehta, R.; Pal, S.; Singh, V.; and Ravi, S. N. 2022. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10422–10431.

Nguyen, Q. P.; Low, B. K. H.; and Jaillet, P. 2020. Variational bayesian unlearning. *Advances in Neural Information Processing Systems*, 33: 16025–16036.

Nguyen, T. T.; Huynh, T. T.; Nguyen, P. L.; Liew, A. W.-C.; Yin, H.; and Nguyen, Q. V. H. 2022. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*.

Parkhi, O.; Vedaldi, A.; and Zisserman, A. 2015. Deep face recognition. In *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*. British Machine Vision Association.

Pawitan, Y. 2001. *In all likelihood: statistical modelling and inference using likelihood*. Oxford University Press.

Sekhari, A.; Acharya, J.; Kamath, G.; and Suresh, A. T. 2021. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34: 18075–18086.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, 3–18. IEEE.

Stephenson, C.; Padhy, S.; Ganesh, A.; Hui, Y.; Tang, H.; and Chung, S. 2021. On the geometry of generalization and memorization in deep neural networks. *arXiv preprint arXiv:2105.14602*.

Tarun, A. K.; Chundawat, V. S.; Mandal, M.; and Kankanhalli, M. 2023a. Deep regression unlearning. In *International Conference on Machine Learning*, 33921–33939. PMLR.

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

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