Can Bad Teaching Induce Forgetting? Unlearning in Deep Networks Using an Incompetent Teacher

Vikram S Chundawat[*1], **Ayush K Tarun**[*1], **Murari Mandal**[2†‡], **Mohan Kankanhalli** [3]

[1]Mavvex Labs, India
[2]School of Computer Engineering, Kalinga Institute of Industrial Technology Bhubaneswar
[3]School of Computing, National University of Singapore
{vikram2000b, ayushtarun210}@gmail.com, murari.mandalfcs@kiit.ac.in, mohan@comp.nus.edu.sg

Abstract

Machine unlearning has become an important area of research due to an increasing need for machine learning (ML) applications to comply with the emerging data privacy regulations. It facilitates the provision for removal of certain set or class of data from an already trained ML model without requiring retraining from scratch. Recently, several efforts have been put in to make unlearning to be effective and efficient. We propose a novel machine unlearning method by exploring the utility of competent and incompetent teachers in a student-teacher framework to induce forgetfulness. The knowledge from the competent and incompetent teachers is selectively transferred to the student to obtain a model that doesn't contain any information about the forget data. We experimentally show that this method generalizes well, is fast and effective. Furthermore, we introduce the *zero retrain forgetting (ZRF) metric* to evaluate any unlearning method. Unlike the existing unlearning metrics, the ZRF score does not depend on the availability of the expensive retrained model. This makes it useful for analysis of the unlearned model after deployment as well. We present results of experiments conducted for random subset forgetting and class forgetting on various deep networks and across different application domains. Source code is at: https://github.com/vikram2000b/bad-teaching-unlearning

Introduction

Machine learning (ML) models are being widely deployed for various applications across different organizations. These models are often trained with large-scale user data. Modern data regulatory frameworks such as European Union GDPR (Voigt and Von dem Bussche 2017), and California Consumer Privacy Act (CCPA) (Goldman 2020) provide for citizens the *right to be forgotten*. It mandates deletion-upon-request of user data. The regulations also require that user consent must be obtained prior to data collection. This consent for the use of an individual's data in these ML models may be withdrawn at any point of time. Thus, a request for data deletion can be made to the ML model owner. The owner company (of the ML model) is legally obligated to remove the models/algorithms derived from using that particular data. As the ML models usually memorize the training samples (Feldman 2020; Carlini et al. 2019), the company either needs to retrain the model from scratch by excluding the requested data or somehow erase the user's information completely from the ML model parameters. The algorithms supporting such information removal are known as *machine unlearning* methods. Machine unlearning also offers a framework to prove data removal from the updated ML model.

The unlearning methods can be practically applied in the following ways: (i) forgetting single-class or multiple classes of data (Tarun et al. 2021), (ii) forgetting a cohort of data from a single class (Golatkar, Achille, and Soatto 2020a,b), (iii) forgetting a random subset of data from multiple classes (Golatkar et al. 2021). In this paper, we investigate the utility of teacher-student framework with knowledge distillation to develop a robust unlearning method that can support all the three modes, i.e. single/multiple class-level, sub-class level and random subset-level unlearning. Another important question we raise is *how well the unlearned model has generalized the forgetting?* Recent studies suggest that the unlearning methods may lead to privacy leakage in the models (Chen et al. 2021). Therefore, it is important to validate whether the unlearned models are susceptible to privacy attacks such as membership inference attacks. Moreover, the trade-off between the amount of unlearning and privacy exposure also should be investigated for better decision-making on the part of the model owner. We propose a new metric to evaluate the generalization ability of the unlearning method.

The existing unlearning methods for deep networks put several constraints over the training procedure. For example, (Golatkar et al. 2021) train an additional mixed-linear model along with the actual model which is used in their unlearning method. Similarly, (Golatkar, Achille, and Soatto 2020a,b) strictly require SGD to be used in optimization during model training. These restrictions and the need for other prior information make these methods less practical for real-world applications. We present a method that does not require any prior information about the training procedure. We

[*]These authors contributed equally.
[†]Work performed while at the School of Computing, National University of Singapore
[‡]Corresponding author

do not train any extra models to assist in the unlearning. Furthermore, we aim to keep the unlearning process efficient and fast in comparison to the high computational costs of the existing methods.

We make the following key contributions:

1. We present a teacher-student framework, consisting of competent and incompetent teachers. The selective knowledge transfer to the student results in the unlearned model. The method works for both single-class and multiple class unlearning. It also works effectively for multiple class random-subset forgetting.
2. We propose a new retrained model-free evaluation metric called zero retrain forgetting (ZRF) metric to robustly evaluate the unlearning method. This also helps in assessing the generalization in the unlearned model on the forget data.
3. Our method works on different modalities of deep networks such as CNN, Vision transformers, and LSTM. Unlike the existing methods, our method doesn't put any constraints over the training procedure. We also demonstrate the wide applicability of our method by conducting experiments in different domains of multimedia applications including image classification, human activity recognition, and epileptic seizure detection.

Related Work

Machine Unlearning. Bourtoule et al. (Bourtoule et al. 2021) proposed to partition the training dataset into non-overlapping shards and create multiple models for the disjoint sets. They store the weakly learned models to deal with multiple data removal requests. Ginart et al. (Ginart et al. 2019) adopted the definition of differential privacy to introduce the probabilistic notion of unlearning. It expects high similarity between the output distributions of the unlearned model and the retrained model without using the deletion data. Several subsequent works (Mirzasoleiman, Karbasi, and Krause 2017; Izzo et al. 2021; Ullah et al. 2021) follow this approach in presenting theoretical guarantees in their respective problem settings. We also follow this definition of unlearning in our work. Guo et al. (Guo et al. 2020) give a certified data removal framework to enable data deletion in linear and logistic regression. Neel et al. (Neel, Roth, and Sharifi-Malvajerdi 2021) apply gradient descent to achieve unlearning in convex models. The difference between differential privacy and machine unlearning is studied in (Sekhari et al. 2021). Unlearning in random forests (Brophy and Lowd 2021) and Bayesian setting (Nguyen, Low, and Jaillet 2020) are also studied. These methods are designed specifically for convex problems and are unlikely to work in deep learning models. Our work is aimed at performing unlearning in deep networks.

Unlearning in Deep Networks. Golatkar et al. (Golatkar, Achille, and Soatto 2020a) presented one of the early works in deep machine unlearning. They introduced a scrubbing method to remove the information from the network weights. The method impose a condition of SGD based optimization during training. The subsequent work (Golatkar, Achille, and Soatto 2020b) proposed a neural tangent kernel (NTK) based method to approximate the training process. The additional approximated model is used to estimate the network weights for the unlearned model. (Golatkar et al. 2021) train a mixed-linear model along with the original model. The linearized model is specific to different deep networks and requires fine-tuning to work properly. Moreover, all these methods suffer from high computational costs, constraints on the training process, and limitations of the approximation methods. Tarun et al. (Tarun et al. 2021) proposed an efficient class-level machine unlearning method. However, it does not support random subset forgetting. In our work, we do not need to train any additional model to support unlearning. Our method does not demand the use of any specific optimization technique during training or any other prior information about the training process. (Chundawat et al. 2023; Graves, Nagisetty, and Ganesh 2021; Tarun et al. 2022) are some other notable works.

Preliminaries

Let the complete (multimedia) dataset be $D_c = \{(x_i, y_i)\}_{i=1}^{n}$ with n number of samples, where x_i is the i^{th} sample, and y_i is the corresponding class label. The set of samples to forget is denoted as D_f. In class-level unlearning, D_f corresponds to all the data samples present in a single or multiple classes. In random-subset unlearning, D_f may either consist of a random subset of data samples from a single class or multiple classes. The information exclusive to these data points need to be removed from the model. The set of remaining samples to be retained is denoted by D_r. The information about these samples are to be kept unchanged in the model. D_f and D_r together represent the whole training set and are mutually exclusive, i.e. $D_r \cup D_f = D_c$ and $D_r \cap D_f = \phi$. Each data point is assigned an unlearning label, l_u, which is 1 if the sample belongs to D_f and 0 if it belongs to D_r. The subset used for unlearning is $\{(x_i, l_{u_i})\}_{i=1}^{p}$, p is total number of samples, and l_{u_i} is unlearning label corresponding to each sample x_i.

The model trained from scratch without observing the forget samples is called the *retrained model* or the *gold model* in this paper. In the proposed teacher-student framework, the *competent teacher* is the fully trained model or the original model. The competent teacher has observed and learned from the complete data D_c. Let $T_s(x;\theta)$ denote the competent/smart teacher with parameters θ. It takes x as input and outputs the probabilities t_s. The *incompetent teacher* is a randomly initialized model. Let $T_d(x;\phi)$ be the incompetent/dumb teacher with parameters ϕ and output probabilities t_d. The student $S(x;\theta)$ is a model initialized with parameters θ i.e., the same as the competent teacher. It returns the output probabilities s. It is to be noted that the student is initialized with all the information present in the original model (θ). The incompetent teacher is used to remove the requested information (about the forget data D_f) from this model. The Kullback-Leibler (KL) divergence (Kullback and Leibler 1951) is used as a measure of similarity between two probability distributions. For two distributions $p(x)$ and $q(x)$, the KL-divergence is defined by

$\mathcal{KL}(p(x)||q(x)) := E_{x\sim p(x)}[log(p(x)/q(x))]$.

Proposed Method

Unlearning with Competent/Incompetent Teachers

We aim to remove the information about the requested datapoints by using two teachers (competent and incompetent) and one student. The student is initialized with knowledge about the complete data i.e., the parameters of the fully trained model. The idea is to *selectively remove the information* about the forget samples from this model. At the same time, the information pertaining to the retain set should not to be disturbed. Thus, the unlearning objective is to remove the information about D_f while retaining the information about D_r. We achieve this by using a pair of (competent/smart (T_s) and incompetent/dumb (T_d)) teachers to manipulate the student (S) as depicted in Figure 1. The bad knowledge about D_f from the incompetent teacher T_d is passed on to the student which helps the student to forget D_f samples. Such an approach consequently induces random knowledge about the forget set in the student instead of completely making their prediction accuracy zero. This serves as a protection against the risk of information exposure about the samples to forget. The bad (random) inputs from T_d may invariably corrupt some of the information about the retain set D_r in the student. Therefore, we selectively borrow correct knowledge related to D_r from the competent teacher T_s as well. In this manner, both the incompetent and competent teachers help the student forget and retain the corresponding information, respectively.

For a student S, incompetent/dumb teacher T_d, and competent/smart teacher T_s, we define the KL-Divergence between T_d and S in Eq. 1.

$$\mathcal{KL}(T_d(x)||S(x)) = \sum_i t_d^{(i)} log(t_d^{(i)}/s^{(i)}) \tag{1}$$

where i corresponds to the data class. Similarly, the KL-Divergence between the fully trained competent teacher T_s and student S is given in Eq. 2.

$$\mathcal{KL}(T_s(x)||S(x)) = \sum_i t_s^{(i)} log(t_s^{(i)}/s^{(i)}) \tag{2}$$

The unlearning objective can be formulated as in Eq. 3.

$$L(x, l_u) = (1 - l_u) * \mathcal{KL}(T_s(x)||S(x)) + l_u * (\mathcal{KL}(T_d(x)||S(x))) \tag{3}$$

where l_u is the unlearning label and x is a data sample. The data samples used by the proposed unlearning method consists of all the samples from D_f and a small subset of samples of D_r. The student is then trained to optimize the loss function L for all these samples. The intuition behind optimizing over L is that we selectively transfer bad knowledge about forget data D_f from T_d by minimizing KL-Divergence between S and T_d and the accurate knowledge corresponding to D_r is fed from T_s by minimizing KL-Divergence between S and T_s. The student learns to mimic T_d for D_f, thus removing information exclusively pertaining to those samples while retaining all the generic information which can be obtained by other samples of same class.

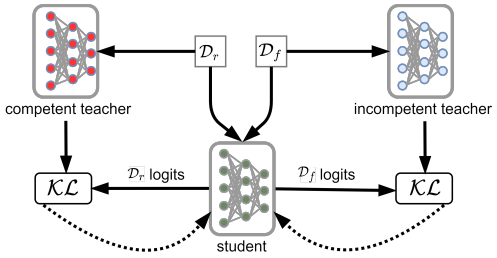


Figure 1: The proposed competent and incompetent teachers based framework for unlearning

Zero Retrain Forgetting Metric

The effectiveness of an unlearning method is evaluated employing several metrics in the literature. Some frequently used metrics are 'accuracy on forget set and retain set' (Golatkar, Achille, and Soatto 2020a; Tarun et al. 2021; Golatkar et al. 2021; Chundawat et al. 2023), relearn time (Tarun et al. 2021), membership inference attacks (Golatkar et al. 2021; Graves, Nagisetty, and Ganesh 2021), activation distance (Golatkar, Achille, and Soatto 2020a; Golatkar et al. 2021), Anamnesis Index (Chundawat et al. 2023), and layer-wise distance (Tarun et al. 2021). Excluding the forget and retain set accuracy, all of the remaining metrics in the literature *require a retrained model* i.e., training a model from scratch without using the forget set. These metrics can only be interpreted with reference to such a retrained model. Such dependency on the retrained model for unlearning evaluation would lead to higher time and computational costs. Simply measuring the performance on D_f and D_r does not reveal whether the information is actually removed from the network weights. Thus it is not a comprehensive measure of unlearning.

We propose a novel 'Zero Retrain Forgetting Metric' (ZRF) to enable evaluation of unlearning methods *free from dependence on the retrained model*. It measures the randomness in the model's prediction by comparing them with the incompetent teacher T_d. We calculate the Jensen–Shannon (JS) divergence(Lin 1991) between an unlearned model M and the incompetent teacher T_d as below.

$$\mathcal{JS}(M(x), T_d(x)) = 0.5 * \mathcal{KL}(M(x)\|m) + 0.5 * \mathcal{KL}(T_d(x)\|m) \tag{4}$$

where $m = \frac{M(x)+T_d(x)}{2}$. The ZRF metric is defined as

$$\mathcal{ZRF} = 1 - \frac{1}{n_f}\sum_{i=0}^{n_f} \mathcal{JS}(M(x_i), T_d(x_i)) \tag{5}$$

where x_i is i^{th} sample from D_f with a total of n_f samples. The ZRF compares the output distribution for the forget set in the unlearned model with the output of a randomly initialized model, which is our incompetent teacher in most of the cases. The ZRF score lies between 0 and 1. The score will be close to 1 if the model behaviour is completely random for the forget samples and it will be close to 0 if the model shows some specific pattern.

What is an ideal ZRF score? Suppose there is a class *aeroplanes* that contains images of *Boeing aircraft* along

with other aircraft models in the training set. If we unlearn *Boeing aircraft*, we don't expect the model to now classify them as *animals*, *vegetables* or any other totally unrelated class. We still expect most of these unlearned images to be classified as aeroplanes. This comes from the intuition that the model must have been designed and trained with generalization in mind. An unlearning method that makes the performance much worse than the generalization error for *aeroplanes* is not actually unlearning. It is just teaching the model to be consistently incorrect when it sees a *Boeing aeroplane*. The ZRF score will be 0 when the model almost always classifies a *Boeing aircraft* as an *animal* or some other totally different class. The ZRF will be 1 if the model always classifies all classes with same random probability for *Boeing aircraft*. Both of these (~ 0 or ~ 1) are not the desirable outcomes. We expect the unlearned model to have a generalization performance similar to that of a model trained without the *Boeing aircraft*. It will have some random predicted logits since the *Boeing aircraft* class was not overfitted during training.

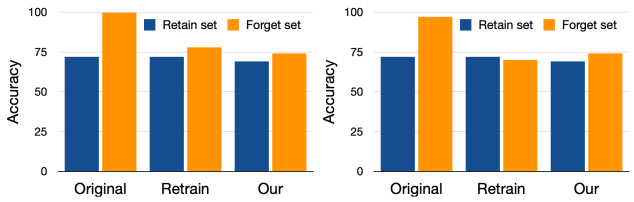
An ideal value of ZRF score depends on the model, dataset and the forget set. Ideally, the optimal ZRF value is what a model trained without the forget set would have. But in practical scenarios we do not have access to the retrained model. So, a good proxy for the ideal ZRF value could be the ZRF value obtained on a test set. The test set by definition is a set about which the model has never learned anything specifically. It is equivalent to saying, a *set* that the model has unlearned perfectly.

Experiments

Datasets used. We evaluate our proposed method on image classification: CIFAR10 (Krizhevsky 2009), CIFAR100 (Krizhevsky 2009), epileptic seizure recognition (Andrzejak et al. 2001), and activity recognition (Anguita et al. 2013) datasets.

Models used. We use ResNet18, ResNet34, MobileNetv2, Vision Transformer, and AllCNN models for learning and unlearning in image classification tasks. We use a 3-layer DNN model for unlearning in epileptic seizure recognition. We use an LSTM model for unlearning in activity recognition task. All the experiments were performed on NVIDIA Tesla V100 (32 GB) with Intel Xeon processors. The experiments are implemented in PyTorch 1.5.0. The KL temperature is set to 1 for all the experiments.

Evaluation Measures. We use the following metrics for our analysis of the proposed unlearning method. *1) Accuracy on forget & retain set:* The accuracy of the unlearned model on D_f and D_r sets should be similar to the retrained model. *2) Membership inference attack:* A membership inference attack is performed to check if any information about the forget samples is still remaining in the model. The attack probabilities should be lower on the forget set in the unlearned model. *3) Activation distance:* This is an average of the L2-distance between the unlearned model and retrained model's predicted probabilities on the forget set. A lesser activation distance represents better unlearning. *4) JS-Divergence:* JS-Divergence between the predictions of the unlearned and retrained model when coupled with activation distance gives a more complete picture on unlearning. Lesser the divergence, better the unlearning. *5) ZRF score:* We introduce this metric to remove the dependence on the retrained model for evaluating the machine unlearning method.



(a) (b)

Figure 2: Unlearning random samples (50 and 100 samples, respectively) from Epileptic Seizure Data.

Baseline Models. We use the unlearning method from (Graves, Nagisetty, and Ganesh 2021) for a comparative analysis. This best fits our problem statement i.e., unlike most other methods, this method achieves unlearning in an already trained model without putting any constraints on the training procedure. We also use the retrained model for comparison. We perform two types of unlearning: (i) sample unlearning, and (ii) class unlearning. We present the experiments and analysis for each of them below.

Forget Acc. Vs Information Exposure Trade-off

Machine unlearning of a specific class or cohort often leads to a decrease in accuracy or performance on forget set. Although, it is an expected result when the forget set is orthogonal to the retain set i.e., there are no samples in the retain set similar to the ones in the forget set. But this may not be true when retain set contains data points similar to the forget set samples. The accuracy may *drop slightly*, *may not drop at all*, or even *increase in some cases*. An unlearning method should bring the forget set performance *closer* to the gold (retrained) model instead of simply reducing it. If the performance of the unlearned model deviates a lot from the gold model, it could lead to *Streisand effect*. This effect refers to unexpected behaviour of the model on forget samples which may leak information about that data. The leak could be in the form of being consistently & maximally incorrect about only the forget samples, signalling that a deliberate effort was made to forget a selected set of samples. The aim should be to avoid this in order to ensure that the information about the forget set has been properly erased. For example, as mentioned earlier in the *aeroplanes* example, when method unlearns *Boeing aircraft*, if it is maximally wrong whenever it sees any *Boeing aircraft* image and classifies it as *sea*, *animals*, *mushroom*, etc., it will be suspicious. Other unseen aeroplanes will not be classified incorrectly so consistently. That means it has not really erased the information of *Boeing aircraft*. That information still exists which the model uses to be deliberately incorrect about the forget set.

Sample Unlearning

CIFARSuper20. The CIFAR100 is made up of 20 super classes i.e., there are different variants for each of these 20

	Super Class	Sub Class	Accuracy ($D_f \downarrow$, $D_r \uparrow$)				ZRF			JS-Div	Mem. Attack Prob		
			Acc.	Orig.	Retrain	**Our**	Orig.	Retrain	**Our**		Orig.	Retrain	**Our**
ResNet18	Veh2	Rocket	D_r D_f	85.78 82	85.79 3	85.05±0.61 2±0.40	0.87	0.93	0.99	0.04	0.98	0.52	0.00
	Veg	MR	D_r D_f	85.82 78	85.38 4	84.79±0.51 1±0.36	0.88	0.93	0.99	0.04	0.99	0.41	0.00
	People	Baby	D_r D_f	85.67 93	85.73 82	85.18±0.54 77±0.34	0.84	0.87	0.98	0.05	1.0	0.84	0.58
	ED	Lamp	D_r D_f	85.83 77	86.28 14	84.74±0.32 5±0.29	0.88	0.94	0.98	0.03	0.98	0.43	0.01
	NS	Sea	D_r D_f	85.63 97	85.46 83	84.58±0.22 84±0.68	0.84	0.87	0.98	0.07	0.99	0.88	0.42
ViT	Veh2	Rocket	D_r D_f	94.89 98	95.35 9	94.84±0.71 17±0.2	0.91	0.96	0.99	0.03	0.99	0.47	0.01
	Veg	MR	D_r D_f	94.94 93	94.8 4	94.59±0.65 17±0.54	0.93	0.98	0.99	0.02	0.99	0.25	0.01
	People	Baby	D_r D_f	94.91 96	95.26 92	94.45±0.69 77±0.33	0.90	0.92	0.99	0.06	1.0	0.91	0.14
	ED	Lamp	D_r D_f	94.93 94	94.86 13	94.99±0.85 21±0.29	0.91	0.96	0.99	0.03	1.0	0.57	0.02
	NS	Sea	D_r D_f	94.91 96	94.93 85	94.31±0.46 79±0.54	0.90	0.92	0.99	0.06	1.0	0.97	0.12

Table 1: Unlearning on CIFARSuper20. We show the results for forgetting a sub-class from a super class. The Original Model is trained on complete dataset. The Retrained Model is trained on retain dataset. We use a randomly initialized model as *incompetent teacher* and the original model as *competent teacher*. The ZRF score should increase on forget set after unlearning. The JS-Div: Jensen-Shannon Divergence, MR: Mushrooms, Acc.: Accuracy, Orig.: Original Model, Veh2: Vehicles2, Veg: Vegetables, ED: Electrical Devices, NS: Natural Scenes

classes. We merge all classes of the CIFAR100 into their super classes and convert it into a 20 class set named CIFARSuper20. Each class in CIFARSuper20 have 5 sub-classes, which are actual classes of CIFAR100. We conduct experiments on CIFARSuper20 by forgetting one sub-class from each super class. This setup makes unlearning more difficult than a regular scenario as we need to unlearn a sample/-class without damaging the information of another sample/-class that looks quite similar to it (for example, forget *baby* from *people* super class consisting of *baby, boy, girl, man, woman*).

We present unlearning results on ResNet18, ResNet34, and Vision Transformer. We use pretrained models to train/fine-tune for 5 epochs using Adam optimizer with a batch size of 256. The learning rate is 0.001 for final layer and 0.0001 for pretrained weight layers. A learning plateau with patience of 3 and reduce factor 0.5 is used. We conduct multiple runs (5 times) of our algorithm which didn't show any significant variation in performance (refer Table 1). Therefore, we report the results of single run for all the models in this paper.

We unlearn various sub-classes from a super-class. We use 30% of retain data and a single epoch of unlearning for all models. The learning rate of 0.0001 is used for unlearning. Table 1 shows unlearning results on ResNet18 and Vision Transformer (Dosovitskiy et al. 2021). The evaluation is performed on all the metrics discussed earlier. It can be observed in Table 1 that performance of our method is very close to that of the retrained model. There is very low probability of membership inference attack on our unlearned model. The accuracy of our method on the forget and retain set when forgetting *rocket* images from *vehicles* is almost same as the retrained models. The membership inference attack probability on the model for samples from *rocket* class drops to 0.002 from 0.982 after unlearning. The JS-Divergence between predictions of the retrained model and our model is 0.04 in the forget set. This implies the output distribution of unlearned model is very close to the retrained model. The ZRF score of our model becomes 0.99 from 0.87 after unlearning, thus indicating effective forgetting. Furthermore, Table 4 shows the unlearning results in ResNet34 with different types of teachers.

Epileptic Seizure Detection. The dataset consists of the status of seizure in medical patients. There are a total of 178 predictor variables and 5 classes. A 3-layer DNN is trained for classification. The model is trained for 50 epochs using Adam Optimizer with learning rate of 0.01 and plateau with patience 10 and reduce factor 0.1. We unlearn 50 and 100 randomly selected data points. The results are presented in Figure 2a and Figure 2b. We observe that the proposed method performance is close to the retrained model. The accuracy on the forget set indicates that we have indeed effectively unlearned the forget set as the forget accuracy is reduced from around 100% to a generalized performance. For example, in case of forgetting 100 samples, the accuracy on the forget set drops from 90% to 74% in our method which is close the 70% accuracy of the retrained model.

Human Activity Recognition. This is a task of classifying the activity of a person using the readings collected from smartphone sensors that an individual is carrying with her.

	Forget Set	Accuracy	Original	Retrain	Amnesiac	**Our**	Activation Distance	
							Amnesiac	**Our**
RN18+ C20 Dataset	Rocket	$D_r \uparrow$	85.78	85.79	84.79	85.05	0.70	**0.67**
		$D_f \downarrow$	82	3	4	2		
	Baby	$D_r \uparrow$	85.67	85.73	84.64	85.18	**0.65**	**0.65**
		$D_f \downarrow$	93	82	78	77		
DNN+ Seizure Dataset	50 samples	$D_r \uparrow$	71.91	72.26	76.04	82.69	0.77	**0.47**
		$D_f \downarrow$	98	70	30	74		
	100 samples	$D_r \uparrow$	71.91	73.39	75.17	79.26	0.69	**0.42**
		$D_f \downarrow$	96	73	40	70		
LSTM+ HAR Dataset	Person #1	$D_r \uparrow$	90.46	84.01	89.45	87.27	0.52	**0.14**
		$D_f \downarrow$	100	99.13	53.03	94.24		
	Person #3	$D_r \uparrow$	90.46	89.68	87.04	86.6	0.49	**0.13**
		$D_f \downarrow$	99.41	99.41	75.95	95.31		

Table 2: Comparison of our method with the Amnesiac learning (Graves, Nagisetty, and Ganesh 2021)
RN18: ResNet18, C20: CIFAR20, HAR: Human Action Recognition

	$\# \mathcal{Y}_f$	Acc.	Orig.	Retrain	UNSIR	**Our**
RN18+ C10 Dataset	1	$D_r \uparrow$	77.86	78.32	71.06	**78.46**
		$D_f \downarrow$	81.01	0	**0**	4.22
	2	$D_r \uparrow$	78.00	79.15	73.61	**79.22**
		$D_f \downarrow$	78.65	0	**0**	9.94
RN18+ Pre+C100 Dataset	1	$D_r \uparrow$	78.68	78.37	75.36	**77.00**
		$D_f \downarrow$	83.00	0	**0**	**0**
	20	$D_r \uparrow$	77.84	79.73	75.38	**77.78**
		$D_f \downarrow$	82.84	0	**0**	3.90
AllCNN+ C10 Dataset	1	$D_r \uparrow$	82.64	85.90	73.90	**81.74**
		$D_f \downarrow$	91.02	0	**0**	9.16
	2	$D_r \uparrow$	84.27	85.21	**80.76**	77.68
		$D_f \downarrow$	79.74	0	**0**	5.64
MNv2+ Pre+C100 Dataset	1	$D_r \uparrow$	77.43	78	75.76	**78.22**
		$D_f \downarrow$	90	0	**0**	**0**
	20	$D_r \uparrow$	76.47	77	76.27	**76.65**
		$D_f \downarrow$	81.70	0	**0**	13.65

Table 3: Class-level unlearning on CIFAR10 and CIFAR100. The results are compared with UNSIR (Tarun et al. 2021). C10: CIFAR10, C100: CIFAR100, RN18: ResNet18, MNv2: MobileNetv2, Pre: Pretrained

The observation were taken from 30 different persons. The dataset contains 6 different types of activities which can be classified using time-series data with sensors giving 9 readings at each time-step. An LSTM Model with 2 dense layers after each LSTM is trained to predict the activity. The model is trained for 50 epochs using Adam Optimizer with learning rate of 0.01 and plateau with patience 10 and reduce factor 0.1. Table 2 contains the results of forgetting person 1 and person 3. Detailed results and effects of various parameters on unlearning are present in the supplementary material.

Comparison with Amnesiac learning (Graves, Nagisetty, and Ganesh 2021). We compare our result with Amnesiac learning which fine-tunes the model with random labels on forget samples. Table 2 shows the comparison between both the methods. We compare the *activation distance* and *accuracy* on the forget and retain set. A lower *activation distance* indicates closeness to the retrained model. This subsequently indicates better unlearning and an accuracy closer to the retrained model is desired on forget and retain set. The *activation distance* for our method is very low compared to Amnesiac method in most of the cases (refer Table 2). Amnesiac method causes too much damage in the forget set of epileptic seizure and human activity recognition dataset, indicating Streisand effect. The accuracy in epileptic seizure dataset (forget set of 50 samples) is 98% for the original model, 70% for retrained model, 74% for our method, and 30% for amnesiac method. The Amnesiac method damages the performance on forget set by a huge margin. It reduces the forget set accuracy to 30% which otherwise should be close to 70%. It should also be noted that *activation distance* from retrained model is 0.47 for our method and 0.77 for Amnesiac method. Amnesiac method is causing undesired effects and the generated model is very different from the retrained model. Our method, besides being more effective and robust, requires access to only a subset of retain data. We use only 30% of retain data to obtain the results in our method. Our method is $\sim 2\times$ faster than Amnesiac method, more effective even when limited data is available for use.

Class Unlearning

We also demonstrate full-class (single and multiple classes) unlearning capability of our method. We show results on CIFAR10 and CIFAR100 with ResNet18, AllCNN, and MobileNetv2 models. Class-level unlearning results are compared with an existing method with configuration as in (Tarun et al. 2021). The model update in our method is performed for 1 epoch using 30% of the retain data. The learning rate at the time of unlearning is set to 0.001. Table 3 gives a performance comparison between the proposed and the existing methods. The accuracy on the retain set in CIFAR10 single-class forgetting is 71.06% for UNSIR, 78.32% for the retrained model, and 78.46% for our method. The results are quite similar in all three methods. The accuracy on forget set is zero in the retrained model and UNSIR but our method retains some accuracy on the forget set. This is because the method learns from a randomly initialized teacher which does random predictions and predicts each class with 10% probability and forget model learns the same. This in turn leads to better protection against the risk of privacy exposure.

T_d	Super Class	Sub Class	Acc.	Orig. Model	Retrain Model	**Our Method**
ResNet34	Veh2	Rocket	$D_r \uparrow$	86.36	85.32	85.8
			$D_f \downarrow$	88	4	1
	Veg	MR	$D_r \uparrow$	86.41	85.92	85.61
			$D_f \downarrow$	83	2	5
ResNet18	Veh2	Rocket	$D_r \uparrow$	86.36	85.32	85.86
			$D_f \downarrow$	88	4	15
	Veg	MR	$D_r \uparrow$	86.41	85.92	85.83
			$D_f \downarrow$	83	2	1
Random	Veh2	Rocket	$D_r \uparrow$	86.36	85.32	86.04
			$D_f \downarrow$	88	4	5
	Veg	MR	$D_r \uparrow$	86.41	85.92	83.37
			$D_f \downarrow$	83	2	11

Table 4: Forgetting sub-class from a super class on CIFAR-Super20+ResNet34 using different types of cheaper incompetent teacher (T_d)

Super-Sub	Acc.	Orig.	Ret.	**(RI)**	**(P)**	JS-Div **(RI)**	JS-Div **(P)**
Veh2-Rocket	$D_r \uparrow$	85.8	85.8	85.1	85.1	0.04	0.02
	$D_f \downarrow$	82	3	2	2		
Veg-MR	$D_r \uparrow$	85.8	85.4	84.8	85.4	0.04	0.08
	$D_f \downarrow$	78	4	1	3		

Table 5: Forgetting sub-class from a super class in CIFARSuper20. Our(RI): Using randomly initialized teacher, Our(P): Using a partially trained model (1 epoch on 50% of retain data) as an incompetent teacher. Ret.: Retrain

Using a Simpler Model as an Incompetent Teacher

Our method does not place any constraints on the architecture of the incompetent teacher. Preferably the architecture should be kept same as the student for proper transfer of information. But it can be replaced with smaller models without significantly affecting the results. As the teacher is initialized with random weights, such behaviour can be obtained by a significantly smaller model, or even hard coded algorithms to generate random predictions. A cheaper teacher can make the unlearning process faster without compromising in the quality of unlearning. We replace the incompetent teacher with (i) a small randomly initialized Neural Network, and (ii) a random prediction generator. The random prediction generator first assigns equal probability to all classes and then a adds Gaussian noise to the predictions. The performance with these teachers is shown in Table 4. With ResNet34 as teacher, the performance on D_f (forget class: *Rocket*) is 1% while using the same model as teacher, 15% while using ResNet18, and 5% while using a random predictor as a teacher. Similarly, the performance on retain set while using ResNet34 as teacher is 85.8%, 85.86% while using ResNet18, and 86.04% while using a random predictor as a teacher. There is a negligible change in performance when we use simpler models as teachers. Thus, it can be used to reduce the computational costs without much loss in the performance.

Using Partially Retrained Model as an Incompetent Teacher

A partially trained (PT) model on a subset of retain data can be used as an incompetent teacher in the proposed framework. Similarly, smaller models trained on a small subset of the retain data can also serve as an incompetent teacher. We show the results of using PT models as incompetent teachers to induce forgetting. Similarly, we further investigate the effectiveness of PT teacher on CIFARSuper20 in Table 5. The teacher is trained for 1 epoch on 50% of the data. The accuracy on forget set *Rocket* is 3% for retrained model, 2% for our method with PT teacher and 2% for RI teacher. The accuracy on retain set for *Rocket* is 85.79% for retrained model, 85.07% for our method with PT teacher and 85.05% for our method with RI teacher. Besides, the JS-Divergence between retrained model & RI model based unlearning is 0.04 and 0.02 in case of PT teacher model based unlearning. This shows that in addition to accuracy improvement, PT teacher based unlearning may also give output distribution more similar to the retrained model.

Efficiency Analysis

We compare the run-time comparison of the retrained model, the existing methods, and the proposed methods. The random weights based setup is $\sim 70\times$ faster than retraining and more than $2\times$ faster than Amnesiac learning (Graves, Nagisetty, and Ganesh 2021). The method is faster when cheaper unlearning teachers are used. The proxy model based setup is about $20\times$ faster than retraining method. The ideal trade-off between efficiency and performance can be obtained by using smaller models partially trained on retain data but they come with the expense of additional training. This (partial training) further comes with a trade-off between computational cost and closeness of the model to the retrained model. The right amount of partial training should be decided. We observed that the cheap randomly initialized models are more efficient and generally perform well in most cases.

Conclusion

We present a novel and general teacher-student framework for machine unlearning. A pair of competent and incompetent teachers is used to selectively transfer knowledge into the student network to obtain the unlearned model. Our work supports single & multiple classes forgetting, sub-class forgetting and random samples forgetting. The effectiveness is evaluated in various application domains and modality of networks. We also introduce a new evaluation metric ZRF that is free from the need of having a retrained model for reference. This metric would be useful in real world scenarios where retrained models are not available or very expensive to obtain. Several possible efficient teachers are also explored to reduce the computational complexity. Future work could focus at the intersection of efficiency and privacy guarantees which may be in the form of either developing better evaluation measures or developing new class of unlearning techniques.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

Andrzejak, R. G.; Lehnertz, K.; Mormann, F.; Rieke, C.; David, P.; and Elger, C. E. 2001. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6): 061907.

Anguita, D.; Ghio, A.; Oneto, L.; Parra Perez, X.; and Reyes Ortiz, J. L. 2013. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, 437–442.

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, 141–159. IEEE.

Brophy, J.; and Lowd, D. 2021. Machine Unlearning for Random Forests. In *International Conference on Machine Learning*, 1092–1104. PMLR.

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.

Chen, M.; Zhang, Z.; Wang, T.; Backes, M.; Humbert, M.; and Zhang, Y. 2021. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 896–911.

Chundawat, V. S.; Tarun, A. K.; Mandal, M.; and Kankanhalli, M. 2023. 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. *ICLR*.

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. Y.; Valiant, G.; and Zou, J. 2019. Making AI Forget You: Data Deletion in Machine Learning. In *Advances in neural information processing systems*, 3513–3526.

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. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European Conference on Computer Vision*, 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. 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. 2020. Certified Data Removal from Machine Learning Models. In *International Conference on Machine Learning*, 3832–3842. PMLR.

Izzo, Z.; Smart, M. A.; Chaudhuri, K.; and Zou, J. 2021. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, 2008–2016. PMLR.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. *Technical report, CIFAR, University of Toronto*.

Kullback, S.; and Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86.

Lin, J. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory*, 37(1): 145–151.

Mirzasoleiman, B.; Karbasi, A.; and Krause, A. 2017. Deletion-robust submodular maximization: Data summarization with "the right to be forgotten". In *International Conference on Machine Learning*, 2449–2458. PMLR.

Neel, S.; Roth, A.; and Sharifi-Malvajerdi, S. 2021. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, 931–962. PMLR.

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

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.

Tarun, A. K.; Chundawat, V. S.; Mandal, M.; and Kankanhalli, M. 2021. Fast Yet Effective Machine Unlearning. *arXiv preprint arXiv:2111.08947*.

Tarun, A. K.; Chundawat, V. S.; Mandal, M.; and Kankanhalli, M. 2022. Deep Regression Unlearning. *arXiv preprint arXiv:2210.08196*.

Ullah, E.; Mai, T.; Rao, A.; Rossi, R. A.; and Arora, R. 2021. Machine unlearning via algorithmic stability. In *Conference on Learning Theory*, 4126–4142. PMLR.

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