- SUPPLEMENTARY -
Learn to Unlearn for Deep Neural Networks: Minimizing Unlearning Interference with Gradient Projection

Tuan Hoang Santu Rana Sunil Gupta Svetha Venkatesh
A2I2, Deakin University
{tuan.h;santu.rana;sunil.gupta;svetha.venkatesh}@deakin.edu.au

1. Experiments

1.1. Membership Inference Attacks

To conduct a membership inference attack (MIA), we train 2 sets of models: 50 original models with full training data and 50 retrained models with only the retaining dataset. We then treat the attack as a binary classification problem on model outputs of the forget dataset, where class **1** represents the samples seen during training (i.e., from original models) and class **0** represents the samples not seen during training (i.e., from the retrained model). For this classification, we use 40 models to extract a training set, 5 models to extract a validation set, and 5 models to extract a test set. The feature for MIA binary classifier comprises the softmax outputs of the models and the one-hot vector representing the sample class-label. We note that for class removal experiment, we remove the row corresponding to the forgetten classes in the MIA features. Subsequently, we train an XGBoost classifier and fine-tune it to achieve the best F1 score on the validation set. Please refer to Figure 7 for more details. We visualize the Receiver Operating Characteristic (ROC) curve and report the Area Under Curve (AUC). The ROC curve illustrates the trade-off between the true-positive rate (TPR) and the false-positive rate (FPR). In their study [1], the authors focus on the extreme scenario of an exceedingly low FPR. They argue that de-identifying even a small number of users within a sensitive dataset is far more important than saying an average-case statement "most people are probably not contained in the sensitive dataset". Nevertheless, in different contexts, such as when potential attackers seek to ascertain whether a user has contributed their data to a model's training, indicating interest in a particular application, the objective may change. Here, the goal is to identify as many data points (and their corresponding users) as possible. Therefore, potentially tolerating a higher FPR to achieve a higher TPR (i.e., Recall) is reasonable.

1.2. Ablation studies

1.2.1 Effect of γ^l

To assess the impact of γ^l on the efficacy of our unlearning approach, we conducted experiments involving the removal of 1,000 samples from the CIFAR-10 dataset (comprising 100 samples per class) using the SmallVGG model. In Table 1, we present the classification error rates for the forgotten dataset $\mathcal{D}_f$ and the retained dataset $\mathcal{D}_r$ as we manipulate the value of γ^l.

Our empirical findings indicate that our method remains robust over a wide range of γ^l values. With a large γ^l (approximately 1, i.e., 0.99 or 0.995), the updating space, i.e., the Residual Gradient Space (RGS), is exceedingly small, hindering the unlearning process. Consequently, the model struggles to eliminate the information associated with the forgotten dataset (lower classification error on forgetting dataset). Conversely, when γ^l is too small (< 0.8), the updating space becomes expansive and contains substantial information about the retainning dataset, this results in a drop in performance on both the retained dataset and the test dataset undesirably. Furthermore, we note that as the updating space is larger, the unlearning process is easier to be over-fit, which could result in the Streisand Effect. An appropriately γ^l facilitates the unlearning process while preserving the information of the retained dataset intact.

1.2.2 Layers

In this section, we conduct experiments to assess the efficacy of our proposed method when only unlearning specific layers. In a manner akin to CFk [2], we freeze the first k layers (the shallower layers), while we exclusively update the deeper layers. More specifically, we conduct two set of experiments: (i) SmallVGG model on CIFAR10 dataset to forget 100 samples per class; and (ii) ResNet-18 model on TinyImageNet dataset to forget the first 5 classes. Experimental results in Table 2 and Table 3 reveal that it is

Table 1. Classification error of the forgetting $\mathcal{D}_f$ and retaining $\mathcal{D}_r$ dataset when forgetting 1000 samples of CIFAR-10 dataset using SmallVGG model at different values of γ^l.

		Classification Error		
		$\mathcal{D}_f$	$\mathcal{D}_r$	$\mathcal{D}_{\text{test}}$
Retrain		9.61	0.00	9.62
γ^l	0.995	1.10	0.00	9.90
	0.99	4.28	0.00	9.88
	0.98	10.04	0.00	9.86
	0.96	10.06	0.00	9.84
	0.90	10.06	0.00	9.84
	0.85	10.04	0.00	9.88
	0.80	13.80	1.74	13.78
	0.75	19.30	8.15	19.14

Table 2. AUC of the MIA ROC with different numbers of unlearned layers of SmallVGG model for GPU and EUk.

Updating top-k	PGU	EUk
0 layer (No-Unlearn)	0.6590	
1 layer	0.6554	0.6587
2 layers	0.5086	0.5587
3 layers	0.5087	0.5379
4 layers	0.5067	0.5168
5 layers (full model)	0.5070	0.5040

possible eliminate information of forgetting dataset retaining in the model outputs by unlearning only a subset of layers instead of the whole models. Yet, the number of layers which needs to update is depended on the architecture and dataset. We will further investigate this problem in future works. In comparison to EUk, simply retraining the last few layers does not effectively remove information of the forgetting dataset on the data removal task. While on the class removal task, retraining the last few layers can be a good approach.

1.2.3 Accuracy losses vs. AUC of MIA ROC

We also analyse the trade-off between accuracy losses in retaining training/testing dataset and the AUC of MIA ROC curve. For the data removal task, from Figure 1 and Figure 3, we can observe that training for more epochs could result in over-fitting for unlearning, which could result in the Streisand Effect. More specifically, Figure 3c shows that when unlearned model is over-fitted, a large number of unlearned samples have very small MIA confidence scores. This could be exploited by attackers to determine membership. Hence, it is important to early stop to avoid over-fitting. For the class-removal task, since we remove the

Table 3. AUC of the MIA ROC with different numbers of unlearned layer blocks of ResNet-18 model for GPU and EUk. For simplicity, we consider the first convolutional layer and the last linear layer as a block.

Updating top-k	PGU	EUk
0 block (No-Unlearn)	0.7053	
1 block	0.7035	0.5350
2 blocks	0.6063	0.5349
3 blocks	0.5409	0.5243
4 blocks	0.5395	0.5198
5 blocks	0.5393	0.5045
6 blocks (full model)	0.5392	0.5045

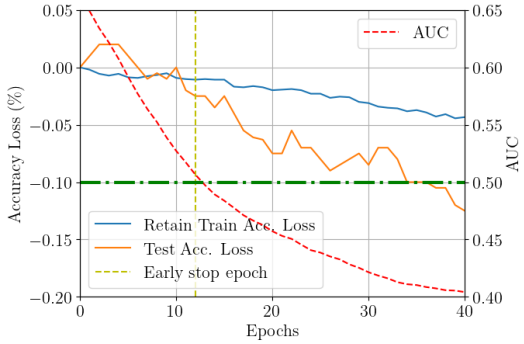


Figure 1. Accuracy loses (compared to the original model) of retaining training and testing dataset (left y-axis) and AUC of MIA ROC curves (right y-axis) by epoch when unlearning 100 smaples per class CIFAR-10 using SmallVGG.

rows corresponding to forgetting classes in the last linear layer of unlearned models, we do not observe the Streisand Effect as in data-removal task. The AUC of ROC curve does not go lower than random level 0.5. Therefore, we can achieve lower AUC of MIA ROC curve at the cost of more losses in the accuracy of the retaining training and testing dataset (and training time). However, it's noteworthy that the AUC remains relatively stable after a certain number of training epochs, e.g., the 70th epoch in Figure 2, indicating that prolonging training beyond this point does not yield discernible benefits. In both experiments, we see that the performance drops of retaining dataset and testing dataset are quite small.

1.2.4 Incremental learning limitation

In this section, we conduct stress tests to analyze the point at which our unlearning method fails, characterized by significant accuracy drops on the test set. We conduct experiments using AllCNN and SmallVGG on CIFAR-10 dataset with the data removal task. We incrementally unlearn 25, 50, or 100 samples per class with the AllCNN model, and 100 samples per class with the SmallVGG model. Figure 5

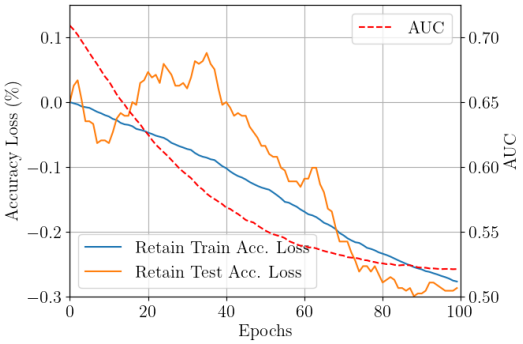


Figure 2. Accuracy loses (compared to the original model) of retaining training and testing dataset (left y-axis) and AUC of MIA ROC curves (right y-axis) by epoch when unlearning 5 classes of TinyImageNet using ResNet-18.

Table 4. Test accuracy for different poisoning budgets for ResNet-18 trained TinyImageNet. The ResNet-18 model achieves 48.18% in test accuracy with *full-clean* training data. We present the accuracy of the model trained with poisoned dataset (***Poisoned***), the model trained with clean data only (exclude the poisoned samples) (***Clean***), and the poisoned model after depoisoning using PGU.

Num. Poison	10K	20K	30K	40K	50K
Poisoned	44.06	39.76	34.42	28.70	22.40
Clean	46.40	45.30	43.90	42.34	39.48
PGU	**45.38**	**44.34**	**43.60**	**42.50**	**41.14**

shows that regardless of step sizes, AllCNN model starts to fail after unlearning 400 samples per class (8% of training size); while SmallVGG model starts to fail at 1100 samples per class (22% of training size). The SmallVGG model fails at much larger number of forgetting samples than AllCNN does potentially because SmallVGG model has more parameters (5.7M) than AllCNN dose (1.6M). This allows SmallVGG to be easily updated to remove information of forgetting dataset. Nevertheless, in both models, we believe that the forgetting dataset sizes are large enough for the life cycle of a model and it is reasonable to retrain the model. Finally, we note that we can easily recover the testing accuracy drops by fine-tuning the unlearned model on retaining dataset; however, in this paper we mainly focus on the setting that the training dataset may be no longer accessible.

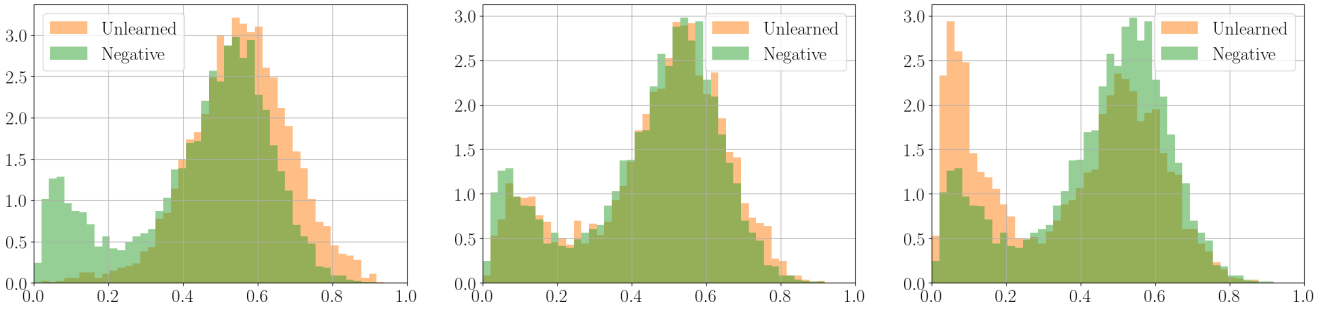
1.3. Depoisoning

We conduct further experiments on the depoisoning setting using ResNet-18 model on TinyImageNet dataset. The experimental results in Table 4 again confirm effectiveness of our method on mitigating the detrimental effects of poisoned samples on a larger model and dataset. Our method can even achieve better test performance than the model that is trained on the clean dataset only when a large number of training dataset is poisoned.

References

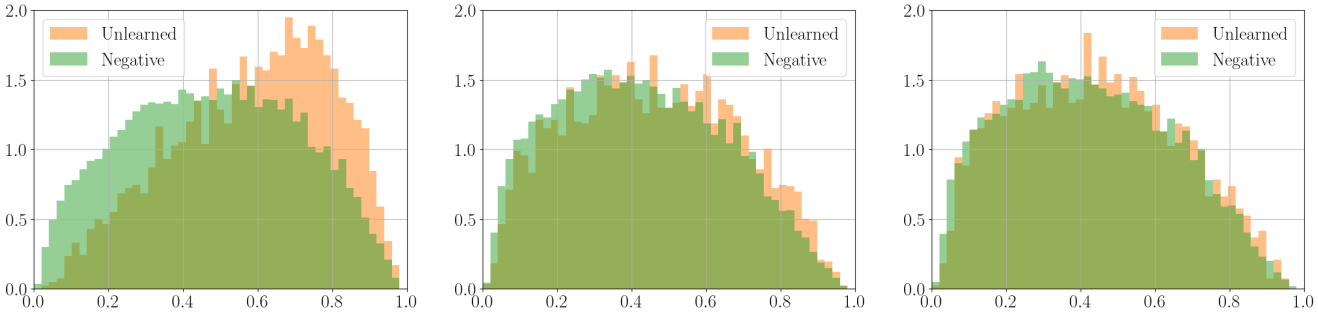
[1] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022. 1

[2] Shashwat Goel, Ameya Prabhu, and Ponnurangam Kumaraguru. Evaluating inexact unlearning requires revisiting forgetting, 01 2022. 1



(a) Epoch 0 (Original) (b) Epoch 12 (Early stop point) (c) Epoch 40

Figure 3. Distributions of the MI attacker confidence scores of unlearned testing set (comprising unlearned and negative samples) during unlearning 100 samples per class of CIFAR-10 using SmallVGG model at different time points: epoch 0 (original), epoch 12 (early stop point) and epoch 40.



(a) Epoch 0 (Original) (b) Epoch 70 (c) Epoch 99

Figure 4. Distributions of the MI attacker confidence scores of unlearned testing set (comprising unlearned and negative samples) during unlearning the first 5-classes of TinyImangeNet dataset using ResNet-18 model at different time points: epoch 0 (original), epoch 70 and epoch 99.

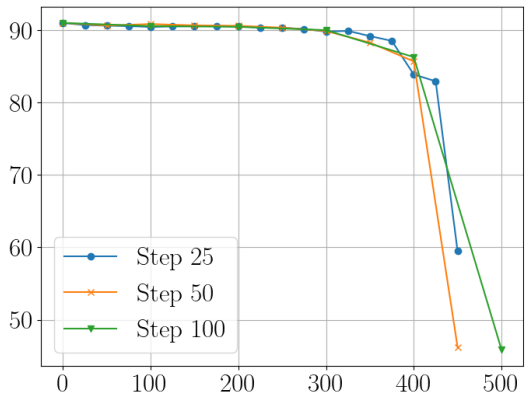


Figure 5. Testing set accuracy of incremental unlearning with step sizes of 25, 50, or 100 samples per class of CIFAR-10 using All-CNN.

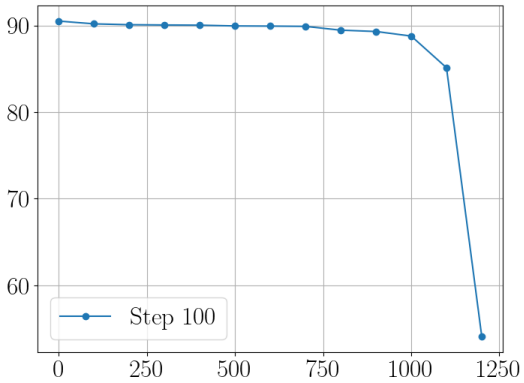


Figure 6. Testing set accuracy of incremental unlearn with step size of 100 samples per class of CIFAR-10 using SmallVGG.

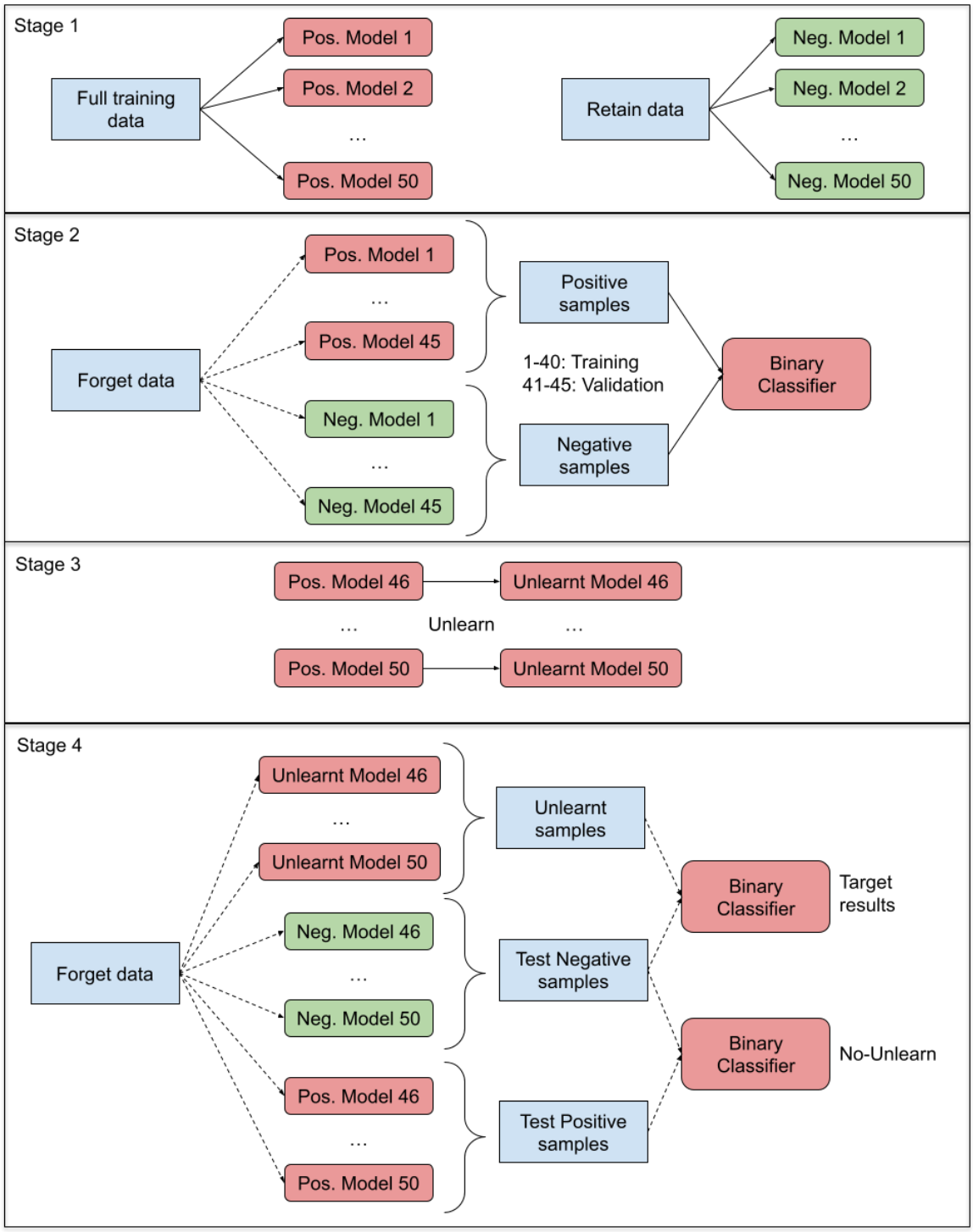


Figure 7. The Membership Inference Attacks (MIA) experiment comprises 4 stages; **Stage 1**: We train 2 sets of models using the full training data (Positive models) and retrain data (Negative models). **Stage 2**: Given the forget data, we collect Positive samples and Negative samples (softmax outputs of the model) from Positive models and Negative models respectively. The Positive and Negative samples are used to train a binary classifier to detect whether a data point is used for training or not. **Stage 3**: From positive models, we apply unlearning methods to obtain unlearnt models. **Stage 4**: Given the forget data, we then extract Test Positive, Test Negative and Unlearnt samples. Test Positive and Test Negative samples are combined to form a balanced no-unlearn test set. Unlearnt and Test Negative samples are combined to form a balance target test set. Note: *Solid arrow lines* indicate training process, *dash arrow lines* indicate inferencing/testing process.