FEW-SHOT UNLEARNING

Youngsik Yoon, Jinhwan Nam, Dongwoo Kim & Jungseul Ok
Department of Computer Science and Engineering,
Pohang University of Science and Technology
{ysyoon97,njh18,dongwookim,jungseul.ok}@postech.ac.kr

ABSTRACT

We consider the problem of machine unlearning to erase target data, which is used in training but incorrect or sensitive, from a trained model while the training dataset is inaccessible. In standard unlearning scenario, it is assumed that the target dataset indicates all the data to be erased. However, this is often infeasible in practice. We hence address a practical scenario of unlearning from a few samples of target data, so-called *few-shot unlearning*. To this end, we devise a new approach employing model inversion to retrieve the training dataset from the trained model. We demonstrate that our method using only a subset of target data outperforms the state-of-the-art methods with a full indication of target data.

1 INTRODUCTION

Unlearning task is to erase a part of the training dataset from a trained model without access to the training dataset. This is useful when we want to correct some of the mislabeled data in training, or to erase specific data for privacy concerns. A standard unlearning scenario is to assume that every target data to be erased is completely indicated (Nguyen et al., 2020; Golatkar et al., 2020; Fu et al., 2021). In practice, however, it is hard to collect such a complete target data. This motivates us to address the problem of *few-shot unlearning*, in which a few examples of the target data are given.

For the few-shot unlearning problem (see Section 2 for a formal formulation), we establish a framework with model inversion (Section 3). To be specific, we first invert the trained model to retrieve the training dataset, and adjusts the model while interpolating the few samples of target data in the retrieved data. This approach provides a direct regularization on the dataset to be retained, while existing methods (Nguyen et al., 2020; Golatkar et al., 2020) use only indirect ones, e.g., regularizing deviation in model parameter space (Nguyen et al., 2020). Our experiment demonstrates that only our approach can perform the few-shot unlearning tasks (Section 4.1), and outperform the existing methods even for the standard unlearning tasks thanks to the direct regularization allowing much larger change of parameters than the indirect ones (Section 4.2).

1.1 RELATED WORK

Machine unlearning. Unlearning problems have been studied in a wide spectrum of assumptions on the accessibility to training and target datasets. Given both datasets, the goal of unlearning is to retrain the model faster than relearning from scratch (Ginart et al., 2019; Bourtoule et al., 2019; Gupta et al., 2021). However, as it is often limited to obtain training dataset after training, a standard setup is to assume that only a trained model and target dataset are given (Fu et al., 2021; Golatkar et al., 2020; Graves et al., 2020; Fu et al., 2021; Nguyen et al., 2020; Tarun et al., 2021; Chundawat et al., 2022; Baumhauer et al., 2020). Since it is unrealistic to indicate all target data to be erased, we further consider a realistic scenario with a limited access to target dataset. Similarly, the zero-shot unlearning problem with no target dataset but target class has been proposed (Tarun et al., 2021; Chundawat et al., 2022), whereas this cannot erase only a (wrong or sensitive) part of class.

Model inversion. We devise a model inversion mechanism to retrieve the training dataset from the trained model but also to interpolate the target dataset from a few examples. As our unlearning scenario possibly includes noisy labels, this is more challenging than model inversion problems in literature: membership attack Shokri et al. (2017), model inversion Fredrikson et al. (2015). We hence fully utilize a set of canonical side information: data augmentation and generative model. This technique itself may be of independent interest for model inversion.

2 PROBLEM FORMULATION

Standard unlearning. We consider a standard classification dataset $\mathcal{D} = \{d_i = (x_i, y_i)\}_{i=1}^N$, where x_i is input data such as an image, and $y_i \in \{1, 2, ..., K\}$ is the corresponding label. Let $f_{w_o}(\cdot) \in [0, 1]^K$ be a classifier of parameter w_o trained on $\mathcal{D}$ with the standard cross-entropy (CE) loss, where a set of input data transformations $\{\phi_\ell\}_{\ell=1}^L$ are used to augment data as a part of enhancing generalization ability, i.e., $w_o = \arg\min_w \sum_{(x_i,y_i)\in D} \sum_\ell \text{CE}(f_w(\phi_\ell(x_i)), y_i)$. We denote D_e and D_r for a partition of dataset D such that D_e is data to be erased and $D_r = D \setminus D_e$ is data to be retained. Then, given w_o and D_e (without D), the standard unlearning task (Nguyen et al., 2020; Golatkar et al., 2020) is to obtain w_u defined as follows:

$$w_u = \arg\min_w \sum_{(x_i,y_i)\in D_r} \sum_\ell \text{CE}(f_w(\phi_\ell(x_i)), y_i) \ , \tag{1}$$

which is the model parameter of the same objective but a different dataset D_r than w. We note that the main challenge of unlearning task is from the absence of access to the remaining data D_r.

Few-shot unlearning. We further consider a practical constraint that only a few-shot of D_e is indicated. More formally, for $\rho \in [0, 1]$, let $D_{e,\rho}$ be the subset of D_e, sub-sampled uniformly at random with ratio ρ such that $\mathbb{E}\left[|D_{e,\rho}|\right] \approx \rho|D_e|$. Then, given w_o and $D_{e,\rho}$, we aim at finding the same unlearning result when we know full D_e, i.e., w_u in equation 1. It is particularly useful to study the case with small ρ. In such cases, we have $D_{r,\rho} := D \setminus D_{e,\rho} \approx D$, and thus the standard unlearning result with $D_{e,\rho}$ should negligibly change from the previous model w_o by definition, c.f., Bayesian (Nguyen et al., 2020) in Figure 1. The few-shot unlearning task needs to address the challenges of the standard unlearning task and, in addition, to interpolate the dataset to be erased from the few-shot $D_{e,\rho}$.

3 PROPOSED METHOD

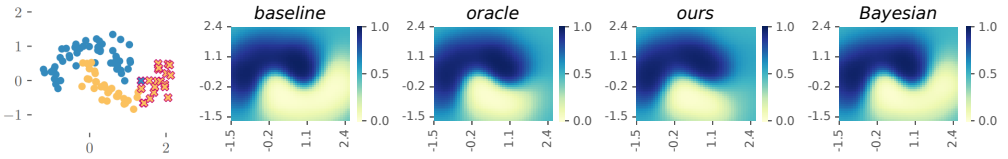
For the few-shot unlearning, we first perform the model inversion to approximate D from w_o. Let D' denote the approximation of D. The standard unlearning can be transformed into a relearning task once one can filter out an interpolation of D_e from D'. We note that our approach overcomes the fundamental limit of previous methods Nguyen et al. (2020); Golatkar et al. (2020), which regularize the deviation in parameter space in order to retain D_r. For instance, Nguyen et al. (2020) assuming Gaussian prior introduces regularizer $\|w_o - w_u\|$ on the deviation from w_o in parametric space. However, this can be problematic when D_e is large, or the behavior of f_w is highly sensitive to w.

For the model inversion, we first observe that if input x is used in training, the output $f_{w_o}(x)$ is concentrated on the one or few classes labeled in the training dataset (which may contain noisy labels). We hence can reveal training data by minimizing the entropy H of $f_{w_o}(x)$ over x. However, the model inversion problem is clearly under-determined. We hence exploit prior knowledge given in generative model G and data augmentation ϕ_ℓ's. We can narrow down searching space via the generative prior G mapping a random noise z of trivial distribution to a point in the approximated domain of D, i.e., searching over z instead of x. In addition, the data augmentation used in training can further introduce a constraint to be verified by a training instance: the consistency of output of the model after the augmentation. However, a target sample may have high entropy due to noisy labeling. We hence employ a classifier to determine whether a sample is from target dataset, or not. To be specific, we approximate the distribution of $D_{e,\rho}$ by multivariate normal distribution on a feature space of w_o, and denote the likelihood of an input x belonging to D_e by $f_e(x)$. In summary, the approximation D' of D is described as follows:

$$D' = \{(G(z), f_{w_o}(G(z))) : H(f_{w_o}(\phi_\ell(G(z)))) < t \ \forall \ell\} \cup \{(G(z), \bar{f}_{w_o}(G(z))) : f_e(G(z)) > t\} \ , \tag{2}$$

where we use the pseudo label $f_{w_o}(G(z))$ for $G(z)$ having low entropy, and a modified pseudo label $\bar{f}_{w_o}$, which has zero at the label of $D_{e,\rho}$, for $G(z)$ similar to inputs of $D_{e,\rho}$.

We generate samples from pre-trained generator G and do data augmentation in given $\{\phi_\ell\}_{\ell=1}^L$. If the entropy of the sample is less than threshold t and the label is consistent on the augmented samples, the sample is added to D'. Moreover, we add $\bar{f}_{w_o}$, the sample with the label softmax of f_{w_o} without the output of unlearned label if the binary classifier classifies the sample as in noisy



(a) Dataset

(b) baseline model and unlearned model results

Figure 1: **Moon classification example** demonstrating the limitation of previous methods on few-shot unlearning setting. Figure 1a visualizes 100 data points of the moon dataset on input dimension $\mathbb{R}^2$, where each point of binary label $y \in \{0, 1\}$ is shown in *blue* if $y = 1$, and *yellow* if $y = 0$.

Table 1: KL divergence of unlearned models against the baseline model and the oracle.

	Ours	EUBO			RKL		
λ	-	10^{-5}	10^{-9}	0	10^{-5}	10^{-9}	0
vs baseline	0.306	0.004	0.012	0.008	0.003	0.007	0.012
vs oracle	0.015	0.394	0.313	0.340	0.432	0.348	0.304

parts. This algorithm can generate an dataset D' approximating the training dataset. We then obtain an approximation of retaining data D_r by deleting D'_e in D' from the binary classifier, where the pseudo code of detailed procedure is described in appendix A.

4 Experiment

4.1 Synthetic dataset

We first perform an unlearning on moon classification task, described in Figure 1a, to compare the existing method built on Bayesian neural network (Nguyen et al., 2020). For a fair comparison, we use the same setting of (Nguyen et al., 2020). However, we aim to unlearn 20 points (red crosses in Figure 1a) of the yellow class when only 5 points are randomly chosen and indicated as few examples of target data, i.e., $\rho = 0.25$. As shown in Figure 1b, our few-show unlearning method (with 5 targets) produces a classifier almost identical to oracle (with 20 targets), while Bayesian unlearning method (Nguyen et al., 2020) fails at the few-shot unlearning.

Table 1 quantifies the comparison of Figure 1 by evaluating KL divergence between the outputs of each unlearning method and reference model (the baseline w_o before unlearning; and the oracle w_u after ideal unlearning) at input points uniformly sampled, i.e., better to have larger divergence to baseline and smaller divergence to oracle. Table 1 clearly shows the superiority of our method over all the variants of the Bayesian unlearning method (Nguyen et al., 2020) with different hyper-parameter λ and submodules: i) evidence upperbound (EUBO); and ii) reverse-KL (RKL).

4.2 Real dataset

We also consider two different unlearning scenarios in image classification with the MNIST dataset: i) we aim to unlearn the model trained with noisy labels in 4.2.1; and ii) we aim to unlearn the model trained with sensitive information in 4.2.2, which needs to be removed. For both scenarios, we use LeNet5 (LeCun et al., 1998) as a backbone model and DCGAN (Radford et al., 2015) pretrained by MNIST as a generator. We use a modified MNIST for each experiment, with two data augmentations: rotating 30 degrees clockwise and counterclockwise. Each experiment is repeated with five random seeds and reported with the mean and standard deviation on unlearning and few-shot unlearning when we can access to all data points in D_e and only 3% ($\rho = 0.03$) of them. We use the same approach for an oracle unlearning in 4.1 and compare our method with the oracle and the Fisher in (Golatkar et al., 2020).

4.2.1 Unlearning noisy label

To simulate the unlearning with the noisy label, we focus on the seven. People write seven in different ways such as 7 and 7̶ (crossed seven), which often confuses the people who are not familiar with the notation. From an observation that a standard classification model often classifies 7̶ as 2 or

Table 2: Accuracy of unlearned models compared among baseline, oracle, ours and Fisher (Golatkar et al., 2020) on the ordinary unlearning and few-shot unlearning setting with the MNIST dataset. $\star$ is the result without augmentation.

(a) Unlearning mislabeled 7

METHOD	ACCURACY	ACCURACY ON 7
BASELINE	96.8 ± 0.45	13.4 ± 0.23
ORACLE	97.9 ± 0.11	89.9 ± 0.03
OURS	97.0 ± 0.29	91.8 ± 0.02
FISHER	97.7 ± 0.05	42.1 ± 5.64
ORACLE(3%)	96.9 ± 0.25	30.8 ± 0.15
OURS(3%)	96.9 ± 0.25	81.1 ± 0.12
FISHER(3%)	97.7 ± 0.00	22.5 ± 1.48
OURS(3%)*	96.1 ± 0.62	19.0 ± 13.2

(b) Unlearning the entire class of 9

METHOD	ACCURACY WITHOUT 9	ACCURACY OF 9
BASELINE	97.9 ± 0.11	96.3 ± 0.92
ORACLE	97.9 ± 0.19	0.0 ± 0.00
OURS	97.4 ± 0.15	0.0 ± 0.00
FISHER	94.4 ± 1.19	88.6 ± 3.05
ORACLE(3%)	97.8 ± 0.19	95.9 ± 1.13
OURS(3%)	97.4 ± 0.24	0.0 ± 0.00
FISHER(3%)	95.0 ± 0.42	89.8 ± 1.74

3 when they learn without 7 , we modify the labels of the MNIST to simulate the label noise. To be specific, we randomly relabel 200 of the 7 in the MNIST into 2, 3, and 7. As a baseline model, we train the model with the relabeled MNIST, and unlearned models are unlearned mislabled 7 from the basline model.

Table 2a shows the accuracy of the models on the full test set and 7 before and after unlearning. After unlearning with the entire data points with incorrect labels, the proposed model outperforms the fisher method on 7 's, while having a small accuracy drop on the full test set. In few-shot unlearning with only 3% of incorrect labels, the proposed method outperforms both the oracle and fisher method by a large margin on 7 's. The results of unlearning without augmentation showed that both the performance of the overall model and the target data were poor. The reason is that there were many cases in which different labels were used when softmax without the output of an unlearned label to suspected of being noisy-labeled data.

4.2.2 UNLEARNING PRIVATE INFORMATION

In the second scenario, we aim to remove an entire class from the trained model under the assumption that the class represents a certain person or private information. To simulate a such scenario, we unlearn the class 9 from the MNIST dataset. From the unlearning, we expect the model accuracy on the class of 9 drops significantly while having a compatible accuracy for the remaining classes. For this experiment, we use the small-sized MNIST that contains the first 10,000 training samples. The baseline classifier is trained on the small MNIST.

Table 2b shows the results of the unlearning. The baseline model has high accuracy in all classes. The fisher method fails to unlearn the class with both the full dataset and a few samples. The oracle achieves zero accuracy on the target class when the full dataset is available and failed to unlearn with only 3% of the dataset. Our method successfully unlearned the target class clearly with the full and partial datasets, while having a small decrease on the other classes.

5 CONCLUSION

We have proposed a few-shot unlearning method to erase a target dataset from a trained model given a few target examples. Our method consists of two parts: inverting model to retrieve training dataset, and relearning with retrieved dataset after excluding target dataset. The model inversion is specialized for unlearning (or noisy label) scenarios, in which we cannot presume that a training example has low entropy on the trained model. We demonstrate that our method using only a subset of target data can achieve similar performance to the state-of-the-art methods with a full indication of target data. We use a pre-trained generative model G, which is possibly unavailable in practice. Hence, it is an interesting future work to consider the case without pre-trained generative prior, in which we need to train a generator from f_{w_o} as in Yoo et al. (2019). In addition, we consider unlearning for classification, while it is also interesting to consider an extension for regression, which seems doable by replacing the entropy with variance for certainty estimation in model inversion.

REFERENCES

Thomas Baumhauer, Pascal Schöttle, and Matthias Zeppelzauer. Machine unlearning: Linear filtration for logit-based classifiers. *arXiv preprint arXiv:2002.02730*, 2020.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. *arXiv preprint arXiv:1912.03817*, 2019.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *arXiv preprint arXiv:2201.05629*, 2022.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

Shaopeng Fu, Fengxiang He, Yue Xu, and Dacheng Tao. Bayesian inference forgetting. *arXiv preprint arXiv:2101.06417*, 2021.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. *arXiv preprint arXiv:2010.10981*, 2020.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. *arXiv preprint arXiv:2106.04378*, 2021.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning. *Advances in Neural Information Processing Systems*, 33, 2020.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE, 2017.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *arXiv preprint arXiv:2111.08947*, 2021.

Jaemin Yoo, Minyong Cho, Taebum Kim, and U Kang. Knowledge extraction with no observable data. 2019.

A PSEUDO CODE OF ALGORITHM

Algorithm 1 Data Generation and Deletion

Input: classifier: c,
generator: G,
unlearn data: $D_{e,p}$,
augmentation: ϕ_1, ϕ_2, ..., ϕ_k
threshold: t, t'
binary classifier: f_e
Output: approximated dataset: D'

$D' \leftarrow []$
Generation
repeat
Sample random noise z.
Generate $X = G(z)$.
Augment X by ϕ_1, ϕ_2, ..., ϕ_k
$m \leftarrow \underset{class}{\text{argmax}}\, c(X)$
if $c(X)[m] \geq t$ and $c(\phi_j(X))[m] \geq t$ for all j **then**
add $(X, c(X))$ to D'
end if
if $f_e(G(z)) > t'$ **then**
$s \leftarrow$ softmax of $c(X)$ without labels of $D_{e,p}$
$m \leftarrow \underset{class}{\text{argmax}}\, s$
if $s[m] \geq t$ **then**
add (X, s) to D'
end if
end if
until num of each class data in $D' == N$
Deletion
repeat
Get data X from D'
if $f_e(G(z)) > t'$ **then**
if $\underset{class}{\text{argmax}}\, c(X) ==$ labels of $D_{e,p}$ **then**
delete X from D'
end if
end if
until visit all data points in D'