UNLEARNING WITH FISHER MASKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning aims to revoke some training data after learning in response to requests from users, model developers, and administrators. Most previous methods are based on direct fine tuning, which may neither remove data completely nor retain full performances on the remain data. In this work, we find that, by first masking some important parameters before fine tuning, the performances of unlearning could be significantly improved. We propose a new masking strategy tailored to unlearning based on Fisher information. Experiments on various datasets and network structures show the effectiveness of the method: without any fine tuning, the proposed Fisher masking could unlearn almost completely while maintaining most of the performance on the remain data. It also exhibits stronger stability comparing with other unlearning baselines.

1 INTRODUCTION

Machine learning algorithms need data for building models. As a large amount of data-driven models rushing into people's daily life, operations on regularizing data usage become crucial. One such operation is removing data from deployed models (also called *machine unlearning* (Cao & Yang, 2015)). For instance, legal laws (e.g., General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA) and Personal Information Protection and Electronic Documents Act (PIPEDA)) declare that users have the right to ask business companies to revoke their personal data. At the same time, models can benefit from removing wrongly annotated data (Rajmadhan et al., 2017; Ren et al., 2021; Pang et al., 2021), systematic biases (Zhao & Chang, 2020; Kim et al., 2019; Serna et al., 2020), and backdoor poisoned data (Chen & Dai, 2021; Yan et al., 2021; Qi et al., 2021).

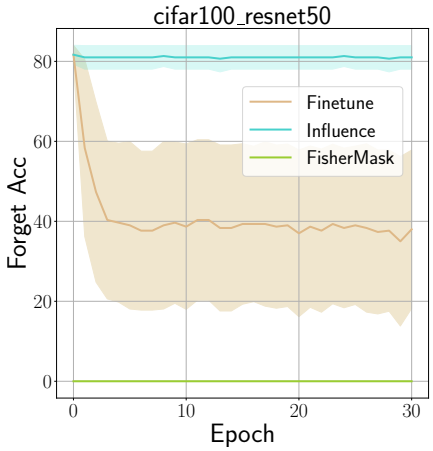

Figure 1: Example unlearning performances. We train ResNet50 with CIFAR-100, and then ask algorithms to remove all pictures from a class. y-axis is the testing performance on that removed class during unlearning. The results indicate that both direct fine tuning and influence function can not effectively unlearn data points. FisherMask is our proposed method.

Given the training set and a subset to remove, the straightforward (and the optimal) way of unlearning is re-learning the model. It guarantees a clean removal, but the computation cost is high. More computationally efficient approaches are based on fine tuning: starting a new learning process on the current model with only remain data. It is known that as the fine tuning process proceeds, the model gradually forgets those unseen data points (*catastrophic forgetting* (Kirkpatrick et al., 2017)). However, fine-tuning-based unlearing could be slow and incomplete in practice. For example, in Figure 1, we ask ResNet50 to remove all pictures belong to one class of CIFAR-100, and after fine tuning on remain samples, it still has about 40% accuracy on the removed class (a clean removal should be 0). Another widely studied fine tuning strategy is based on second-order optimization. Koh & Liang (2017); Guo et al. (2020) show that with one-step Newton update (also called *influence function*), the new model's prediction behaviour correlates well with the re-learned model, but the strong correlation there doesn't imply a successful unlearning: in Figure 1, the one-step Newton

update almost remove no information (80% accuracy). Hence, given the special initial state (parameters), unlearning with fine tuning is still a challenge: it is hard to escape the local optimum of the old model.

In this work, we study methods for accelerate unlearning by adding proper perturbations on its initial state. Instead of adding them randomly (as suggested by most Hitchhiker's guides to escape local optimum), we would like the perturbations are biased towards the task of removing data. To accomplish this, we first identify parameters which are important for modeling the excluded data, and mask them before fine tuning the model. Our main finding is that Fisher information plays a key role in masking parameters: it characterizes how parameters contribute to the distance between models before and after unlearning. We develop a new masking strategy based on Fisher information which shows strong unlearning performances across different datasets and deep neural network structures. We conduct extensive empirical evaluations on fine-tuning-based unlearning methods with fair and reproducible configurations The main empirical findings are,

- comparing with direct fine tuning, the masking strategy significantly improve unlearning performances. In fact, even with random masking, the unlearning process could be accelerated.
- comparing with other parameter saliency scores (e.g., neuron activation scores and gradient-based scores Sundararajan et al. (2017)), masking with Fisher information is effective on balancing removing and reserving. Without any fine tuning, Fisher masking could unlearn almost completely while maintaining most of the performance on the remain data.
- unlearning algorithms could be unstable with respect to different network structures, datasets, and learning rate settings. The Fisher masking exhibits the best stability among current fine-tuning-based methods.

2 BACKGROUND

Given a dataset $D = \{(x_i, y_i)\}_{i=1}^{|D|}$, $y_i \in \mathcal{Y}$ is the label of input $x_i \in \mathcal{X}$. A learning algorithm tries to minimize the loss function $\mathcal{L}(w, D) = \sum_{i=1}^{N} \ell(x_i, y_i, w)$ on the training set D, where w is the model parameter, and ℓ is the log-loss $\ell(x, y, w) = -\log p(y|x, w)$. We denote $w^* = \arg\min_w \mathcal{L}(w, D)$.

Let D_f be the subset that we want to remove from the model (*forget set*), and $D_r = D \setminus D_f$ contains remain data samples (*remain set*). D_f could be any subset of D, but for evaluation purpose, we mainly focus on cases that D_f contains all samples belong to the same class (Shibata et al., 2021; Wang et al., 2022). In this case, the target of unlearning is to obtain $\hat{w}_r$ which 1) has similar performances on D_r as $w_r^* = \arg\min_w \mathcal{L}(w, D_r)$, and 2) contains no information about D_f (i.e., zero accuracy on samples in D_f like w_r^*). [1]

Instead of fully re-training from scratch, one could solve the objective $\arg\min_w \mathcal{L}(w, D_r)$ from w^* (i.e., fine tuning). While any optimization procedure could be applied (e.g., SGD), Koh & Liang (2017); Koh et al. (2019); Golatkar et al. (2020a) show that for the special setting of the initial state ($w_0 = w^*$), one-step Newton update could be an effective move towards the unlearning target w_r^*. Specifically, the influence function used in (Koh & Liang, 2017; Koh et al., 2019) approximates the difference between w^* and w_r^* with,

$$w_r^* \approx \hat{w}_r = w^* + \frac{1}{|D_f|} \nabla_w^2 \mathcal{L}(w^*, D)^{-1} \nabla_w \mathcal{L}(w^*, D_f).$$

The approximation is obtained by Taylor expansion of $\mathcal{L}(w, D)$ at the stationary w^*, which assumes a small forget set D_f. Golatkar et al. (2020a) add a noise term to the one-step Newton update which aims to approximately minimize KL-divergence between $\hat{w}_r$ and w_r^*,

$$w_r^* \approx \hat{w}_r = w^* - \nabla_w^2 \mathcal{L}(w^*, D_r)^{-1} \nabla_w \mathcal{L}(w^*, D_r) + (\lambda \sigma^2)^{\frac{1}{4}} \nabla_w^2 \mathcal{L}(w^*, D_r)^{-\frac{1}{4}} \epsilon, \tag{1}$$

where λ, σ are hyperparameters, and $\epsilon \sim N(0, I)$ is a Gaussian noise.

Although fine tuning with Newton-updates only needs one step, it is expensive to compute Hessian matrices for deep neural networks. Koh & Liang (2017) apply the iterative LiSSA algorithm (Agarwal

[1] For arbitrary D_f, zero accuracy is not sufficient for validating unlearning. For example, if we have $D_f = D_r$ (duplicate datasets), the unlearned model should have identical behaviours on D_f and D_r. We will evaluate the performance of arbitrary D_f in the task of denoise (Table 2).

et al., 2017) to approximate the Hessian. Golatkar et al. (2020a) simply drop the second term of Equation 1, and approximate Hessian with diagonals of Fisher matrix in the noise term,

$$\hat{w}_r = w^* + (\lambda\sigma^2)^{\frac{1}{4}} h^{-\frac{1}{4}}, \tag{2}$$

where vector h contains diagonal entries of the Fisher matrix computed on w^* for dataset D_r.

3 UNLEARNING APPROACHES

In this section, we first show that Fisher information is important for identifying key parameters for unlearning, and based on this observation, we propose a new masking strategy (`FisherMask`). We then describe two alternative masking methods (`ActivationMask` and `GradMask`) in Section 3.2 and 3.3. We also introduce a setting of learning rates for the following fine tuning process which makes unlearning stable in practice.

3.1 MASKING WITH FISHER INFORMATION

For a distribution $p(y|x, w)$, Fisher matrix (and its empirical estimation) is defined by

$$F \triangleq \mathrm{E}_{x,y}\left[\nabla_w \log p(y|x,w) \nabla \log p(y|x,w)^T\right] \approx \frac{1}{|D|}\sum_{i=1}^{|D|} \nabla_w \log p(y_i|x_i,w) \nabla \log p(y_i|x_i,w)^T.$$

For large-scale neural networks, it is usually expensive to use full F, thus we will further approximate F with its diagonal $\mathrm{diag}(F)$ following (Kirkpatrick et al., 2017; Golatkar et al., 2020a). It is known that F equals to negative expectation of $\log p(y|x, w)$'s Hessian,

$$F = -\mathrm{E}_{x,y} \nabla_w^2 \log p(y|x,w).$$

We now take linear regression as an example to show the role of Fisher information in unlearning. Let $p(y|x, w) = \frac{1}{Z}\exp\{-\frac{1}{2}(w^T x - y)^2\}$, and $X = [x_1, x_2, \ldots, x_{|D|}]$. The empirical Fisher now is $F = |D|^{-1} X X^T$. Let $F_{jj} = \sum_{x_i \in D} x_{ij}^2$, be the diagonals of the Fisher matrix (to simplify notations the factor $|D|$ is dropped), where x_{ij} is the j-th dimension of x_i. Let $F_{r,jj} = \sum_{x_i \in D_r} x_{ij}^2$ and $F_{f,jj} = \sum_{x_i \in D_f} x_{ij}^2$ be the remain set and forget set's contribution to the j-th diagonal of the Fisher matrix.

Let M be the set of parameters to be masked, and $\hat{w}_r$ be the parameter obtained by masking w^* with M, whose j-th entry is $\hat{w}_{r,j} = \begin{cases} w_j^*, & j \notin M \\ 0, & j \in M \end{cases}$.

Proposition 1. *For linear regression, if we approximate Fisher matrix F with its diagonals $\mathrm{diag}(F)$ and assume all diagonals are restrict positive, the KL-divergence between the optimal model w_r^* and the masked model $\hat{w}_r$ has the following upper bound,*

$$\mathrm{KL}(w_r^*, \hat{w}_r) \leq \frac{\lambda}{2|D|}\left(c + 2c_1 \sum_{j \notin M} \frac{1}{F_{jj}^2}\left(\frac{F_{f,jj}}{F_{r,jj}}\right)^2\right), \tag{3}$$

where λ is the largest eigenvalue of XX^T, and c, c_1 are constants depend on the remain set D_r.

The upper bound implies that to make the masked parameter $\hat{w}_r$ close to the unlearning target w_r^*, the unmasked set $\overline{M}$ should contain those parameters with small Fisher information contribution on the forget set $F_{f,jj}$ and large Fisher information contribution on the remain set $F_{r,jj}$, which means the masking strategy should do the opposite. Therefore, we develop `FisherMask` strategy to select top R parameters according to $F_{f,jj} - F_{r,jj}$ as M.

Proposition 1 could be extended to generalized linear models: the proof depends the close form solution of linear regression, while similar estimation of solutions could be established for generalized linear models (Yang et al., 2015). We also remark that, though `FisherMask` performs quite well for deep models (e.g., models with parameterized representation layers), we now don't obtain a similar upper bound for them.

3.2 MASKING WITH ACTIVATION VALUES

In neural networks, an alternative way to measure importance of parameters is inspecting activation states of their corresponding neurons. As Erhan et al. (2009) suggested, maximizing a neuron's activation value with respect to input could be a good first-order representation of what the neuron is doing. Here, we could find neurons which maximizes the activation on the forget set D_f and mask corresponding parameters to get perturbations on the old model.

Suppose we have a trained L layer CNN model. First, for each training sample i, we average activation scores of an output channel j (obtained by Conv-BatchNorm-ReLu operations on an intermediate input channel), and record the score to A_{ij} of a table A. Then, for each channel, we can compute its averaged activation values $A_{D_r,j} = \frac{1}{|D_r|}\sum_{i\in D_r} A_{ij}$ over the remain set D_r, and similarly, $A_{D_f,j}$ on the forget set D_f. The `ActivationMask` strategy identifies top R of channels with large $A_{D_f,j} - A_{D_r,j}$, and mask CNN kernel parameters connecting with those channels.

Wang et al. (2022) propose an improved version of `ActivationMask` which not only looks at how a channel activates, but also how it contributes to the whole activation pattern of the entire class (their method can only remove all samples of a class). Specifically, their `TF-IDF` method masks neurons with term-frequency inverse-document-frequency scores, which analogize channels to words and classes to documents in information retrieval. It is worth a mention that, the activation value there is calculated before going through BatchNorm layer, which means information stored in BatchNorm layers can not be removed.

3.3 MASKING WITH GRADIENT INFORMATION

Besides forward information used in `ActivationMask`, we can also consider backward (gradient) information for finding important unlearning parameters. Here, we use Integrate Gradient (Sundararajan et al., 2017) which are widely applied (Dai et al., 2022; Hao et al., 2021). Given a CNN model, like `ActivationMask`, we obtain table A containing activation value of the i-th sample at the j-th (hidden) output channel. The importance of j-th channel is evaluated by how a small perturbation on j affects the final loss, which can be approximated by $B_{ij} = A_{ij} \times \int_{\alpha=0}^{1} \frac{\partial \mathcal{L}(x_i,w)}{\partial A_{ij}} \mathrm{d}\alpha$ [2]. We can also compute $B_{D_r,j} = \frac{1}{|D_r|}\sum_{i\in D_r} B_{ij}$ and $B_{D_f,j} = \frac{1}{|D_f|}\sum_{i\in D_f} B_{ij}$ to characterize contribution of D_r and D_f to j. However, since it is more expensive to compute A_{ij} comparing with `ActivationMask`. The masking with gradient information `GradMask` only selects neurons according to $B_{D_f,j}$ ($|D_r| \gg |D_f|$ in most cases).

3.4 FINE TUNING

After masking, we fine tune parameters on the D_r to recover the performance on remain data. We find that the final unlearning performances could be sensitive to different settings of learning rate, which are usually ignored in current unlearning configurations. For example, Wang et al. (2022) chooses a fixed learning rate 0.1 in fine tuning process (and in training), but constant learning rate is not the standard setting of modern optimization algorithms.

In experiments, we deploy a learning rate scheduler for unlearning to mimics the original learning process in a shorter period (denoted by S, and we set $S = 5$). For example, if the original learning process triggers a decay of rate at $1/2$ of learning epochs, then the unlearning process also performs the same decay at the $S/2$. We find that the replay of learning rate scheduling makes unlearning more stable.

4 EXPERIMENT

Remove Full Category To evaluate unlearning strategies, we mainly focus on removing a full category of samples. We remove the first category on all datasets. The target unlearned model should have zero accuracy on the unlearn class during testing.

[2] The integration is approximated by Riemman approximation following (Sundararajan et al., 2017).

Remove Outliers Different with removing a full class of samples, evaluating removing a group of random data points is challenge since w^* and w_r^* could be quite close. To compare the effectiveness of unlearning methods on random group, we construct synthetic datasets by shuffling labels within the random group, which we refer as outliers or noise. We test performances of models after unlearning those outliers.

Experimental Setups We evaluate our unlearning methods with various combinations of networks and datasets. We experiment on 4 datasets and 4 networks which results in 16 models. Datasets we choose include CIFAR10/100 (Krizhevsky et al., 2009), MNIST and Tiny-ImageNet (Le & Yang, 2015). And networks include ResNet (He et al., 2016), VGG (Simonyan & Zisserman, 2015), GoogLeNet (Szegedy et al., 2015) and DenseNet (Huang et al., 2017). As previous studies (Ma et al. 2021; Le & Hua 2021) point out, different learning settings, especially the small learning rate and insufficient training epochs, could lead to different results in network pruning. On CIFAR10/100 and Tiny-ImageNet, we train 160 epochs and the learning rate decrease by a factor of 0.1 after 80 and 120 epochs with initial learning rate 0.1, following Ma et al. (2021). On MNIST, we train model at fixed learning rate 0.01 for 30 epochs. Detailed dataset statistics and experiment setups are in Appendix 4. We set the parameter mask ratio R=0.04 for dataset MNIST and Tiny-ImageNet, and R=0.02 for dataset CIFAR10/100 for all mask methods.

We compare different unlearning masking strategies (`FisherMask` in Section 3.1, `ActivationMask` in Section 3.2, and `GradMask` in Section 3.3) with following baselines [3]:

- `Finetune`, directly fine tuning model w^* on the remain data D_r with same optimizer of the learning process.
- `RandomMask`, randomly masking parameters with same ratio and then fine tuning on D_r.
- `FisherNoise`, unlearning method proposed in (Golatkar et al., 2020a) which adds fisher noise to destroy the weights that may have been informative about D_r (Equation 2). [4]
- `TF-IDF`, unlearning method proposed in (Wang et al., 2022). which uses TF-IDF score to select parameters and then fine tunes on the dataset D_r.

For unlearning a full category, we compare these methods on two parts of the test set: test samples belong to the unlearn category and other samples (forget set and remain set, shortly). To distinguish from the data in the training set, we will indicate them as the remain training data and forget training data. We also show the results of different readout functions in Appendix C.

4.1 Main Results

Figure 2 shows the performances of different removing mechanisms on test set. We list the results of ResNet20 on CIFAR10, GoogLeNet on CIFAR100, VGG16 on MNIST and DenseNet on Tiny-ImageNet, and the results of the remain settings can be found in Appendix D. Accuracies on forget and remain sets of different methods are pictured in the fist and second row, respectively. From the results, we can find that,

- First, the retrained model takes a long time to learn, which indicates the necessity of unlearning methods. Comparing with the retrained models (red dashed curves), all unlearning strategies could accelerate the learning process and achieve a comparable performance to the final performance of retrained model (red horizontal dashed lines).
- Second, unlearning only with fine tuning (`Finetune` and `FisherNoise`) could be not enough. The `Finetune` method could not unlearn completely on most settings, while `FisherNoise` method unlearns too much even including the critical information for remain data. For example, on the dataset CIFAR10/100 and Tiny-ImageNet, `Finetune` method still remains high accuracy on forget set, and `FisherNoise` method has a poor remain accuracy on all datasets, especially on CIFAR100 and MNIST datasets. Moreover, even masking random parameters (`RandomMask`) helps unlearning: it has better forget results compared to `Finetune` method. It may because that randomly mask parameters helps optimizing on the new loss of the fine tuning process, and makes it easier to find a better local optimum.

[3] All experiments are conducted on a single 2.5GHz core and a single NVIDIA GTX 3090 GPU.

[4] Hyper-parameters are set as in (Golatkar et al., 2020a).

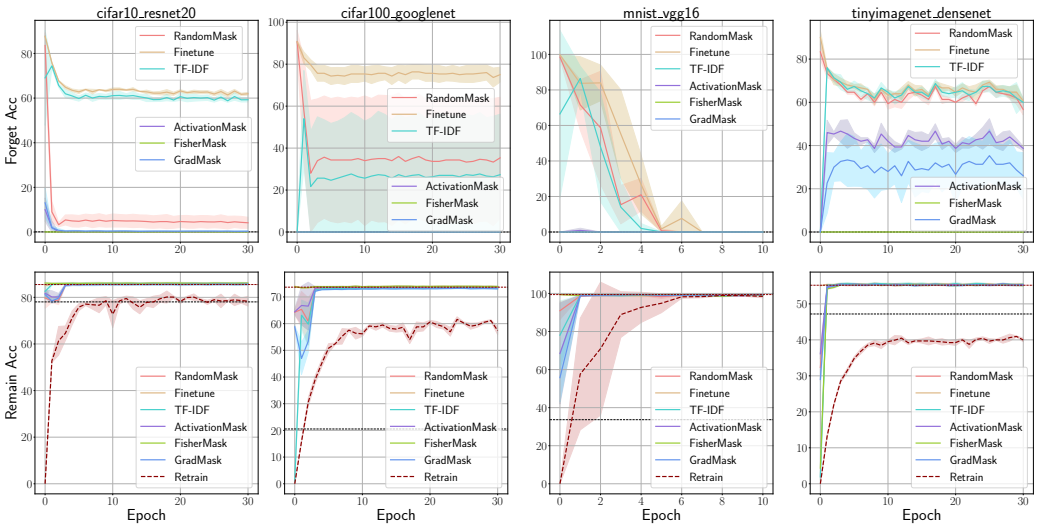

Figure 2: Results of unlearning on test sets. We show the change of accuracy during the fine tuning process. All results are averaged over 3 runs. (A) **performances on forget set** are listed in the first raw, the lower is the better. (B) **performances on remain set** are listed in the second raw, the closer to the model w_r^* is the better. The red horizontal dashed line indicates the final performance of model w_r^*, and the black horizontal dashed line indicates the `FisherNoise` method.

Average	remain		forget	
	Acc	Volatility	Acc	Volatility
Finetune	**74.18**±19.39	1.82	85.28±20.47	0.27
RandomMask	65.75±24.36	0.02	83.80±16.93	-63.84
TF-IDF	42.23±38.95	33.6	38.83±42.44	0.71
FisherNoise	36.62±28.24	-	**0.26**±0.94	-
ActivationMask	63.10±30.31	12.7	4.4±10.54	0.17
GradMask	57.08±24.72	18.68	7.56±19.14	0.2
FisherMask	65.05±36.29	10.91	1.42±6.06	0.01

Table 1: Performance averaged on all model settings (16 models * 3 runs) after applying removing mechanisms. We use the volatility metric to measure the degree of change in the model performance curve during fine tuning.

- Third, `FisherMask` method performs best when unlearning with only masking without fine tuning. We list the average performances of different removing mechanisms on all experiment settings without fine tuning process in Table 1. The results of `Finetune` method indicate the performance of original trained model w^*. As we can see from the results, `FisherMask` method can unlearn efficiently and still retain a high performance on remain data compared to other unlearn methods. It maintains 87% of its original performance while almost forgot completely. Although `FisherNoise` can almost perfectly remove information, it also removes too much useful information and has the lowest remain accuracy among all methods.

- Finally, when unlearning with both masking and fine tuning, it is not sufficient to only see performances on remain sets (even if the unlearning target is to approximate w_r^*). It is because there may be many local optima that are good in the performance of remain set but cannot unlearn completely. Considering both performances on remain and forget set, `ActivationMask`, `GradMask` and `FisherMask` obtain appreciable increase in performances. Among these methods, `FisherMask` method not only unlearn completely on all experiment settings, but also exhibits the best stability among other methods, which shows the effectiveness of Fisher information in finding the key parameters. `ActivationMask` performs comparably in most settings except DenseNet on Tiny-ImageNet. Considering a faster running time, `ActivationMask` can be good choice of unlearning method in most settings. `GradMask` performances relatively poor, probably because we only mask parameters in in convolution layer and information in the BatchNorm layer can not be touched.

Criterion	10%		30%		50%	
	Test Acc	Noise Acc	Test Acc	Noise Acc	Test Acc	Noise Acc
Finetune	83.04±0.36	81.42±1.55	80.05±1.03	79.95±1.17	76.91±0.08	77.44±0.14
RandomMask	83.22±0.47	81.93 ±1.29	80.16±0.68	80.19±1.01	77.23±0.49	77.26±0.18
ActivationMask	83.06±0.31	81.49 ±1.72	80.16±0.71	79.96±0.81	77.18±0.17	77.26 ±0.10
GradMask	82.62±0.29	82.00±1.51	80.33±0.76	**80.26**±1.11	77.14±0.02	77.23±0.08
FisherMask	**83.25**±0.22	**82.13**±1.12	**80.36**±0.71	80.22±1.01	**77.45**±0.40	**77.59**±0.02

Table 2: Outlier deletion experiment on CIFAR10 dataset with ResNet20. We present the accuracy on test set and noisy training data points with corrected labels. The test and noise acc of the model without noisy training data are 84.43 ± 0.20 and 84.67 ± 0.35.

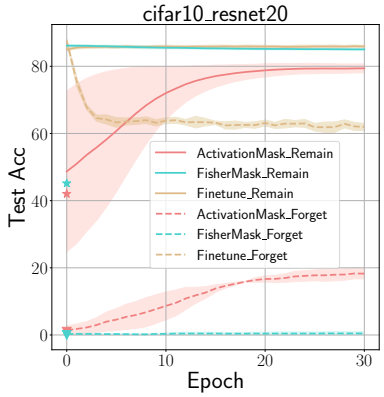

Figure 3: Performances of `ActivationMask` and `FisherMask` with limited fine-tuning data. We use ★ and ▼ to indicate the performances of only forget data on the remain and forget set, respectively. And the curves shows the performances with limited remain data (0.1%).

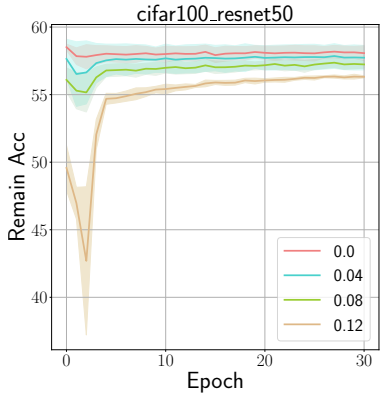

Figure 4: Performances on remain data for `FisherMask` method with different removing ratios. `FisherMask` could unlearn completely under all removing ratios, but as the remove ratios become larger, the performance on remain data degrades.

We notice that some performance curves in the Figure 2 fluctuate severly, such as the forget accuracy of `TF-IDF` method on Tiny-ImageNet with DenseNet: forgetting performances becomes 0 after removing, but they grows rapidly after few fine tuning epochs. We calculate a volatility score for curves to measure stability of the different methods. The volatility scores are calculated as $\frac{1}{|T|-1}\sum_{t=1}^{|T|}(\text{Acc}_t - \text{Acc}_{t-1})$ and $\frac{1}{|T|-1}\sum_{t=0}^{|T|-1}(\text{Acc}_t - \text{Acc}_{t+1})$ for remain data and forget data (Acc_t is the testing accuracy of the model at t-th epoch). The results show that all masking method have smaller volatility score on forget set than baseline methods, and `FisherMask` has the smallest score which means its performance does not change rapidly. On the remain set, only the `Finetune` and `RandomMask` have a low volatility score. Except the two, `FisherMask` still have a smallest score which also demonstrates a good stability in the fine tuning process.

4.2 Outlier Deletion

We conduct outlier deletion experiment on CIFAR10 with ResNet20. We randomly shuffle labels the training points to create outliers (with different noisy ratios), and then remove them with various removing mechanisms. The `FisherNoise` and `TF-IDF` method are not included for they are designed to remove categories. We fine-tune model for T epochs, then list the best accuracy.

Results are shown in Table 2. We report testing accuracy both on the test set and on the removed data points with corrected labels (i.e., their original labels), which characterize performances on unseen data and ability to correct wrong predictions. The results show that all three methods performs better than baseline methods in most settings, and `FisherMask` method performs best both on test set and corrected noisy data. With the increase of noise ratio, `FisherMask` performs better than other methods more obviously.

4.3 WITH LIMITED REMAIN DATA

Here, we first consider the situation that the whole dataset could not be fetched and only the forget training set is available. In the previous experiments, we use the whole dataset for `ActivationMask` and `FisherMask` to find parameters to be masked. Here, we only have forget data for scoring parameters, and don't run fine tuning. We test the two models with different experiment settings (full results are in Appendix E), the results on CIFAR10 dataset with ResNet20 model are shown in Figure 3 (marked as ★ and ▼).

Next, we consider an easier scenario where we can get a small portion of the remain data instead of the whole set. We randomly sample 50 samples (full remain dataset contains 45000 samples, 0.1%) from remain training data. We use these data both in parameter masking and fine tuning process. The performances of unlearning are depicted in Figure 3 (presented in curves).

From the results, we can see that: first, when no remain training data provided, both `ActivationMask` and `FisherMask` method can unlearn efficiently, but `ActivationMask` has a lower remain accuracy compared to `FisherMask`; second, with only 0.1% of the remain training data provided, (i) `Finetune` still remains a high forget accuracy which shows the difficulty to complete unlearning; (ii) `ActivationMask` has an unstable forget accuracy during the fine tuning process. Its forgetting curve of gradually increase, while in Figure 2, the forget accuracy are always 0 with full remain training set provided. It indicates that limited remain training data could affect the calculation on the importance score of the parameters for `ActivationMask`. Regarding remain accuracy, it also fails to recover the fully performances; (iii) `FisherMask` can keep a stable forget performance while fully recover the remain accuracy. The remain accuracy could be boosted from 53% (45.16 / 85.02) to almost identical (86.26 / 85.02) with only 0.1% of the remain data. It may suggest that importance scores derived from Fisher information helps to improve data efficiency of unlearning.

4.4 DIFFERENT REMOVAL RATIOS

Here we show the performance for `FisherMask` method with different remove ratios on remain dataset in Figure 4. Accuracy on forget dataset remains 0 as remove ratio ranges from 0 to 0.12, but accuracy on the remain set changes a lot. Performances degrade significantly as the percentage of masking increases. Besides that, oscillation of curves also becomes progressively larger as the increase of remove ratio. When the remove ratio is relatively small, the performance change curve is relatively flat. It will have a small drop of performances (because the learning rate is a bit large at the start) and quickly pick up. However, the performance drops a lot when the remove ratio is higher. Therefore, when we scrub too many information, it is hard for fine tuning to find them back even with the full remain training set.

4.5 LEARNING RATE SCHEDULING

Considering that usually the learning rate starts from a large value and decays slowly during epochs, how to choose a appropriate learning rate in our fine tuning process could be a problem. A large learning rate could be helped for accelerating learning process, while a small learning rate helps to approach local minima and get better performance. As we want to recover the performance of remain data as quickly as possible (as few fine tuning epochs as possible), we compress the scheduler of the original learning rate to the first S epochs (Section 3.4). We show the results of different learning rate on CIFAR10 with ResNet20 model in Figure 5.

It can be seen that compared to using a constant learning rate [5]: (i) for `Finetune`, a large learning rate is helpful for forgetting, but may lead to a worse and less stable remain performance. A smaller learning rate helps keeping high remain accuracy but is harmful to the forgetting performance. Our scheduler could find a balance between stable high remain accuracy and low forget accuracy. (ii) for `ActivationMask`, the results are similar. our scheduler could recover the performance of the remain data as quickly as possible (as small learning rate) while maintaining stability and accelerating unlearning (as large learning rate). (iii) for `FisherMask`, it is less influenced by the learning rate.

[5] 0.1, 0.01 and 0.001 are the learning rates at the initial time, after the first decay and after the second decay, respectively.

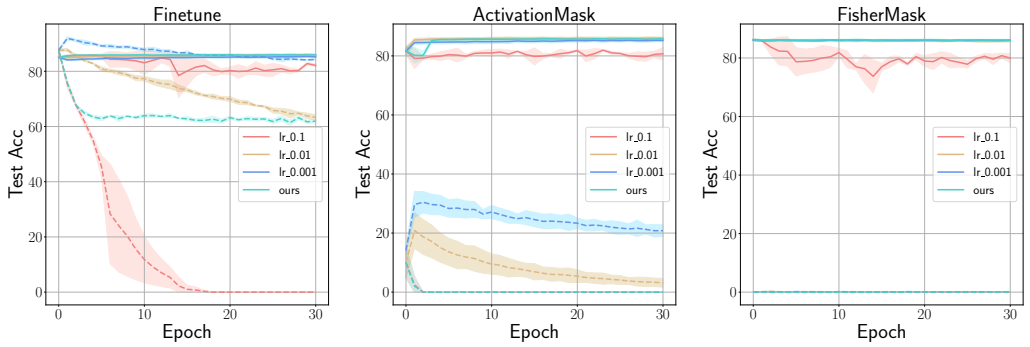

Figure 5: Experiment results of different learning rate schedule used in fine tuning process on CIFAR10 dataset with ResNet20. The solid and dashed lines indicate the performance on the remain and forget datasets, respectively.

5 Related Work

Machine unlearning, first proposed by Cao & Yang (2015) in the context of statistical query learning, aims to forget training samples for data protection and model security. Most of the previous works in machine unlearning focus on linear models, need to calculate the reverse of hessian matrix, and perform a single SGD update towards the minimizer of the approximation (Koh & Liang 2017; Guo et al. 2020; Izzo et al. 2021). Besides linear models, Ginart et al. (2019) investigate an effective data deletion algorithm for the specific setting of k-means clustering. Brophy & Lowd (2021) apply a variant of random forests that enables the removal of training data with minimal retraining. For deep neural networks, Golatkar et al. (2020a) try to add a fisher noise to hide the information about unlearn data. The work closest to ours is (Wang et al., 2022) (which is a concurrent work). They try to scrub memories for each category in federated learning. but unlike our method, they calculate activation maps on the dataset in each layer and use TF-IDF to choose neurons after grouping and averaging activation maps by category. After pruning, there also utilize a fine-tuning process to recover the performance.

Contrary to machine unlearning, life-long learning or continual learning, is often viewed as the concept to learn many tasks sequentially without forgetting the knowledge obtained from preceding tasks. The term "forgetting" mentioned here is *Catastrophic Forgetting* (French, 1993), which results in model overfitting on the currently available data and suffering from performance deterioration on the previously trained data. However, Golatkar et al. (2020a) show that finetune on the remain dataset from the original trained model could not suffer catastrophic forgetting, while our experiments present different results which we attribute to the different learning settings. A lot of works have done to constraint forgetting, such as, regularization-based methods (Kirkpatrick et al. 2017; Aljundi et al. 2018) propose to selectively slow down the learning rate of task important parameters; rehearsal-based methods (Chaudhry et al. 2019; Hayes et al. 2019) save a data buffer to recover performance of old data when new task comes; and architecture-based methods (Li et al. 2019; Loo et al. 2021) have separate components for each task. Our work based on the previous found (Bau et al., 2020) that one subset of neurons can be highly activated by specific training images, which motivates us to separate the parameters for unlearn data and remain data.

6 Conclusion

In this paper, we study different masking strategies to accelerate unlearning. We find our masking strategies significantly improve unlearning performances and exhibits a better stability among other methods. Experiments on various architectures and datasets show that all our methods performs better than baselines and `FisherMask` method performs best while `ActivationMask` method could achieve a good performance with a fast running speed. Future work will explore reducing the fine-tune time for our methods.

7 REPRODUCIBILITY STATEMENT

To ensure that our experimental results are reproducible, we run three random seeds for all experiments and inscribe the mean and variance on the performance curves, for example Figure 2 and 3. What's more, to ensure we get the generic conclusions among datasets and structures, we run our unlearning strategies comparing with baseline methods on 4 common datasets and 4 common architectures. The details of the dataset statistics and model training setups can be found in Appendix B. Our code is provided in the supplementary.

REFERENCES

Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *Journal of Machine Learning Research*, 18(116):1–40, 2017. URL http://jmlr.org/papers/v18/16-491.html.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11207 of *Lecture Notes in Computer Science*, pp. 144–161. Springer, 2018. doi: 10.1007/978-3-030-01219-9_9. URL https://doi.org/10.1007/978-3-030-01219-9_9.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Àgata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proc. Natl. Acad. Sci. USA*, 117(48):30071–30078, 2020. doi: 10.1073/pnas.1907375117. URL https://doi.org/10.1073/pnas.1907375117.

Jonathan Brophy and Daniel Lowd. Machine unlearning for random forests. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1092–1104. PMLR, 2021. URL http://proceedings.mlr.press/v139/brophy21a.html.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pp. 463–480. IEEE Computer Society, 2015. doi: 10.1109/SP.2015.35. URL https://doi.org/10.1109/SP.2015.35.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Chuanshuai Chen and Jiazhu Dai. Mitigating backdoor attacks in lstm-based text classification systems by backdoor keyword identification. *Neurocomputing*, 452:253–262, 2021. doi: 10.1016/j.neucom.2021.04.105. URL https://doi.org/10.1016/j.neucom.2021.04.105.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 8493–8502. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.581. URL https://doi.org/10.18653/v1/2022.acl-long.581.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Robert M. French. Catastrophic interference in connectionist networks: Can it be predicted, can it be prevented? In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector (eds.), *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pp. 1176–1177. Morgan Kaufmann, 1993. URL http://papers.nips.cc/paper/799-catastrophic-interference-in-connectionist-networks-can-it-be-predicted-can-it-be-prevented.

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 3513–3526, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/cb79f8fa58b91d3af6c9c991f63962d3-Abstract.html.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 9301–9309. Computer Vision Foundation / IEEE, 2020a. doi: 10.1109/CVPR42600.2020.00932. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Golatkar_Eternal_Sunshine_of_the_Spotless_Net_Selective_Forgetting_in_Deep_CVPR_2020_paper.html.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX*, volume 12374 of *Lecture Notes in Computer Science*, pp. 383–398. Springer, 2020b. doi: 10.1007/978-3-030-58526-6_23. URL https://doi.org/10.1007/978-3-030-58526-6_23.

Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842. PMLR, 2020. URL http://proceedings.mlr.press/v119/guo20c.html.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 12963–12971. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17533.

Tyler L. Hayes, Nathan D. Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pp. 9769–9776. IEEE, 2019. doi: 10.1109/ICRA.2019.8793982. URL https://doi.org/10.1109/ICRA.2019.8793982.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 2261–2269. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.243. URL https://doi.org/10.1109/CVPR.2017.243.

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In Arindam Banerjee and Kenji Fukumizu (eds.), *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2008–2016. PMLR, 2021. URL http://proceedings.mlr.press/v130/izzo21a.html.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 9012–9020. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00922. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Kim_Learning_Not_to_Learn_Training_Deep_Neural_Networks_With_Biased_CVPR_2019_paper.html.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 2017. URL http://proceedings.mlr.press/v70/koh17a.html.

Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 5255–5265, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/a78482ce76496fcf49085f2190e675b4-Abstract.html.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Duong H. Le and Binh-Son Hua. Network pruning that matters: A case study on retraining variants. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=Cb54AMqHQFP.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3925–3934. PMLR, 2019. URL http://proceedings.mlr.press/v97/li19m.html.

Noel Loo, Siddharth Swaroop, and Richard E. Turner. Generalized variational continual learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=_IM-AfFhna9.

Xiaolong Ma, Geng Yuan, Xuan Shen, Tianlong Chen, Xuxi Chen, Xiaohan Chen, Ning Liu, Minghai Qin, Sijia Liu, Zhangyang Wang, and Yanzhi Wang. Sanity checks for lottery tickets: Does your winning ticket really win the jackpot? In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 12749–12760, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6a130f1dc6f0c829f874e92e5458dced-Abstract.html.

Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-recorrupted: Unsupervised deep learning for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 2043–2052. Computer Vision Foundation / IEEE, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Pang_Recorrupted-to-Recorrupted_Unsupervised_Deep_Learning_for_Image_Denoising_CVPR_2021_paper.html.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 443–453. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.37. URL https://doi.org/10.18653/v1/2021.acl-long.37.

Ekambaram Rajmadhan, Dmitry B. Goldgof, and Lawrence O. Hall. Finding label noise examples in large scale datasets. In *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017, Banff, AB, Canada, October 5-8, 2017*, pp. 2420–2424. IEEE, 2017. doi: 10.1109/SMC.2017.8122985. URL https://doi.org/10.1109/SMC.2017.8122985.

Chao Ren, Xiaohai He, Chuncheng Wang, and Zhibo Zhao. Adaptive consistency prior based deep network for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 8596–8606. Computer Vision Foundation / IEEE, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Ren_Adaptive_Consistency_Prior_Based_Deep_Network_for_Image_Denoising_CVPR_2021_paper.html.

Ignacio Serna, Alejandro Peña, Aythami Morales, and Julian Fiérrez. Insidebias: Measuring bias in deep networks and application to face gender biometrics. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pp. 3720–3727. IEEE, 2020. doi: 10.1109/ICPR48806.2021.9412443. URL https://doi.org/10.1109/ICPR48806.2021.9412443.

Takashi Shibata, Go Irie, Daiki Ikami, and Yu Mitsuzumi. Learning with selective forgetting. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 989–996. ijcai.org, 2021. doi: 10.24963/ijcai.2021/137. URL https://doi.org/10.24963/ijcai.2021/137.

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 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 3–18. IEEE Computer Society, 2017. doi: 10.1109/SP.2017.41. URL https://doi.org/10.1109/SP.2017.41.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017. URL http://proceedings.mlr.press/v70/sundararajan17a.html.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 1–9. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298594. URL https://doi.org/10.1109/CVPR.2015.7298594.

Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (eds.), *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pp. 622–632. ACM, 2022. doi: 10.1145/3485447.3512222. URL https://doi.org/10.1145/3485447.3512222.

Zhicong Yan, Gaolei Li, Yuan TIan, Jun Wu, Shenghong Li, Mingzhe Chen, and H. Vincent Poor. Dehib: Deep hidden backdoor attack on semi-supervised learning via adversarial perturbation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 10585–10593. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17266.

Eunho Yang, Aurelie C Lozano, and Pradeep K Ravikumar. Closed-form estimators for high-dimensional generalized linear models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/17d63b1625c816c22647a73e1482372b-Paper.pdf.

Jieyu Zhao and Kai-Wei Chang. LOGAN: local group bias detection by clustering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 1968–1977. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.155. URL https://doi.org/10.18653/v1/2020.emnlp-main.155.