Privacy-preserving Collaborative Learning with Automatic Transformation Search

Wei Gao[1,2], Shangwei Guo[3], Tianwei Zhang*[1], Han Qiu[4], Yonggang Wen[1], Yang Liu[1]
[1]Nanyang Technological University, [2]S-Lab, Nanyang Technological University,
[3]Chongqing University, [4]Tsinghua University
{gaow0007,tianwei.zhang,ygwen,yangliu}@ntu.edu.sg,swguo@cqu.edu.cn,qiuhan@tsinghua.edu.cn

Abstract

Collaborative learning has gained great popularity due to its benefit of data privacy protection: participants can jointly train a Deep Learning model without sharing their training sets. However, recent works discovered that an adversary can fully recover the sensitive training samples from the shared gradients. Such reconstruction attacks pose severe threats to collaborative learning. Hence, effective mitigation solutions are urgently desired.

In this paper, we propose to leverage data augmentation to defeat reconstruction attacks: by preprocessing sensitive images with carefully-selected transformation policies, it becomes infeasible for the adversary to extract any useful information from the corresponding gradients. We design a novel search method to automatically discover qualified policies. We adopt two new metrics to quantify the impacts of transformations on data privacy and model usability, which can significantly accelerate the search speed. Comprehensive evaluations demonstrate that the policies discovered by our method can defeat existing reconstruction attacks in collaborative learning, with high efficiency and negligible impact on the model performance.

1. Introduction

A collaborative learning system enables multiple participants to jointly train a shared Deep Learning (DL) model for a common artificial intelligence task [36, 22, 9]. Typical collaborative systems are distributed systems such as federated learning systems, where each participant iteratively calculates the local gradients based on his own training dataset and shares them with other participants to approach the ideal model. This collaborative mode can significantly improve the training speed, model performance and generalization. Besides, it can also protect the training data privacy, as participants do not need to release their sensitive data during the training phase. Due to these advantages, collaborative learning has become promising in many scenarios, e.g., smart manufacturing [10], autonomous driving [25], digital health [3], etc.

Although each participant does not disclose the training dataset, he has to share with others the gradients, which can leak information of the sensitive data indirectly. Past works [14, 22, 23] demonstrated the possibility of membership inference and property inference attacks in collaborative learning. A more serious threat is the *reconstruction attack* [40, 38, 6], where an adversary can recover the exact values of samples from the shared gradients with high fidelity. This attack is very practical under realistic and complex circumstances (e.g., large-size images, batch training).

Due to the severity of this threat, an effective and practical defense solution is desirable to protect the privacy of collaborative learning. Common privacy-aware solutions [40, 33] attempt to increase the difficulty of input reconstruction by obfuscating the gradients. However, the obfuscation magnitude is bounded by the performance requirement of the DL task: a large-scale obfuscation can hide the input information, but also impair the model accuracy. The effectiveness of various techniques (e.g., noise injection, model pruning) against reconstruction attacks have been empirically evaluated [40]. Unfortunately, they cannot achieve a satisfactory trade-off between data privacy and model usability, and hence become less practical.

Motivated by the limitations of existing solutions, this paper aims to solve the privacy issue from a different perspective: *obfuscating the training data to make the reconstruction difficult or infeasible.* The key insight of our strategy is to *repurpose data augmentation techniques for privacy enhancement*. A variety of transformation approaches have been designed to improve the model performance and generalization. We aim to leverage certain transformation functions to preprocess the training sets and then train the gradients, which can prevent malicious participants from reconstructing the transformed or original samples.

Mitigating reconstruction attacks via data augmentation

*Corresponding author.

is challenging. First, existing image transformation functions are mainly used for performance and generalization improvement. It is unknown which ones are effective in reducing information leakage. Second, conventional approaches apply these transformations to augment the training sets, where original data samples are still kept for model training. This is relatively easier to maintain the model performance. In contrast, to achieve our goal, we have to abandon the original samples, and only use the transformed ones for training, which can impair the model accuracy.

We introduce a systematic approach to overcome these challenges. Our goal is to automatically discover an ensemble of effective transformations from a large collection of commonly-used data augmentation functions. This ensemble is then formed as a transformation policy, to preserve the privacy of collaborative learning. Due to the large search space and training overhead, it is computationally infeasible to evaluate the privacy and performance impacts of each possible policy. Instead, we design two novel metrics to quantify the policies without training a complete model. These metrics with our new search algorithm can identify the optimal policies within 2.5 GPU hours.

The identified transformation policies exhibit great capability of preserving the privacy while maintaining the model performance. They also enjoy the following properties: (1) the policies are general and able to defeat different variants of reconstruction attacks. (2) The input transformations are performed locally without modifying the training pipeline. They are applicable to any collaborative learning systems and algorithms. (3) The transformations are lightweight with negligible impact on the training efficiency. (4) The policies have high transferability: the optimal policy searched from one dataset can be directly applied to other datasets as well.

2. Related Work

2.1. Reconstruction Attacks

In collaborative learning, reconstruction attacks aim to recover the training samples from the shared gradients. Zhu et al. [40] first formulated this attack as an optimization process: the adversarial participant searches for the optimal samples in the input space that can best match the gradients. L-BFGS [21] was adopted to realize this attack. Following this work, several improved attacks were further proposed to enhance the attack effects and reduce the cost. For instance, Zhao et al. [38] first extracted the training labels from the gradients, and then recovered the training samples with higher convergence speed. Geiping et al. [6] adopted the cosine similarity as the distance function and Adam as the optimizer to solve the optimization problem, which can yield more precise reconstruction results. He et al. [12, 13] proposed reconstruction attacks against collaborative inference systems, which are not considered in this paper.

2.2. Existing Defenses and Limitations

One straightforward defense strategy is to obfuscate the gradients before releasing them, in order to make the reconstruction difficult or infeasible. Differential privacy is a theoretical framework to guide the randomization of exchange information [1, 20, 27, 8]. For instance, Zhu et al. [40] tried to add Gaussian/Laplacian noises guided by differential privacy to the gradients, and compress the model with gradient pruning. Unfortunately, there exists an unsolvable conflict between privacy and usability in these solutions: a large-scale obfuscation can compromise the model performance while a small-scale obfuscation still leaks certain amount of information. Such ineffectiveness of these methods was validated in [40], and will be further confirmed in this paper (Table 3). Wei et al. [33] proposed to adjust the hyper-parameters (e.g. bath size, loss or distance function), which also has limited impact on the attack results.

An alternative direction is to design new collaborative learning systems to thwart the reconstruction attacks. Zhao et al. [39] proposed a framework that transfers sensitive samples to public ones with privacy protection, based on which the participants can collaboratively update their local models with noise-preserving labels. Fan et al. [5] designed a secret polarization network for each participant to produce secret losses and calculate the gradients. These approaches require all participants to follow the new training pipelines or optimization methods. They cannot be directly applied to existing collaborative implementations. This significantly restricts their practicality.

3. Problem Statement

3.1. System Model

We consider a standard collaborative learning system where all participants jointly train a global model M. Each participant owns a private dataset D. Let $\mathcal{L}, W$ be the loss function and the parameters of M, respectively. At each iteration, every participant randomly selects a training sample (x, y), calculates the loss $\mathcal{L}(x, y)$ by forward propagation and then the gradient $\bigtriangledown W(x, y) = \frac{\partial \mathcal{L}(x,y)}{\partial W}$ using backward propagation. The participants can also use the mini-batch SGD, where a mini-batch of samples are randomly selected to train the gradient at each iteration.

Gradients need to be consolidated at each iteration. In a centralized system, a parameter server aggregates all the gradients, and sends the updated one to each participant. In a decentralized system, each participant aggregates the gradients from his neighbors, and then broadcasts the results.

3.2. Attack Model

We consider a honest-but-curious adversarial entity in the collaborative learning system, who receives other participants' gradients in each iteration, and tries to reconstruct the private training samples from them. In the centralized mode, this adversary is the parameter server, while in the decentralized mode, the adversary can be an arbitrary participant.

Common reconstruction techniques [40, 38, 6] adopt different optimization algorithms to extract training samples from the gradients. Specifically, given a gradient $\bigtriangledown W(x, y)$, the attack goal is to discover a pair of sample and label (x', y'), such that the corresponding gradient $\bigtriangledown W(x', y')$ is very close to $\bigtriangledown W$. This can be formulated as an optimization problem of minimizing the objective:

$$x^*, y^* = \operatorname*{argmin}_{x', y'} \quad || \bigtriangledown W(x, y) - \bigtriangledown W(x', y')||, \qquad (1)$$

where $|| \cdot ||$ is a norm for measuring the distance between the two gradients. A reconstruction attack succeeds if the identified x^* is visually similar to x. This can be quantified by the metric of Peak Signal-to-Noise Ratio (PSNR) [15]. Formally, a reconstruction attack is defined as below:

Definition 1. *((ϵ, δ)-Reconstruction Attack) Let (x^*, y^*) be the solution to Equation 1, and (x, y) be the target training sample that produces $\bigtriangledown W(x, y)$. This process is called a (ϵ, δ)-reconstruction attack if the following property is held:*

$$\Pr[\texttt{PSNR}(x^*, x) \geq \epsilon] \geq 1 - \delta. \qquad (2)$$

4. Methodology

4.1. Overview

Driven by the severity of reconstruction attacks, and limitations of existing defenses, we focus on a new mitigation opportunity in this paper: *transforming the sensitive training samples to make the reconstruction difficult or even infeasible*. Image transformation has been widely adopted to mitigate adversarial examples [29, 30, 31], backdoor attacks [37], and attack watermarking schemes [7]. We repurpose it for defeating reconstruction attacks. Specifically, given a private dataset D, we aim to find a policy composed of a set of transformation functions $T = t_1 \circ t_2 \circ ... \circ t_n$, to convert each sample $x \in D$ to $\widehat{x} = T(x)$ and establish a new dataset $\widehat{D}$. The data owner can use $\widehat{D}$ to calculate the gradients and safely share them with untrusted collaborators in collaborative learning. Such a transformation policy must satisfy two requirements: (1) the adversarial participant is not able to infer $\widehat{x}$ (and x) from $\bigtriangledown W(\widehat{x}, y)$. (2) The final model should maintain similar performance as the one trained from D. We formally define our strategy as below:

Definition 2. *($(\epsilon, \delta, \gamma)$-Privacy-aware Transformation Policy) Given a dataset D, and an ensemble of transformations T, let $\widehat{D}$ be another dataset transformed from D with T. Let M and $\widehat{M}$ be the models trained over D and $\widehat{D}$, respectively. T is defined to be (ϵ, δ, γ)-privacy-aware, if the following two requirements are met:*

$$\Pr[\texttt{PSNR}(x^*, \widehat{x}) < \epsilon] \geq 1 - \delta, \forall x \in D, \qquad (3)$$

$$\texttt{ACC}(M) - \texttt{ACC}(\widehat{M}) < \gamma, \qquad (4)$$

where $\widehat{x} = T(x)$, x^* is the reconstructed input from $\bigtriangledown W(\widehat{x}, y)$, and ACC *is the prediction accuracy function.*

It is critical to identify the transformation functions that can satisfy the above two requirements. With the advance of computer vision, different image transformations have been designed for better data augmentation. We aim to repurpose some of these data augmentation approaches to enhance the privacy of collaborative learning.

Due to the large quantity and variety of augmentation functions, we introduce a systematic and automatic method to search for the most privacy-preserving and efficient policy. Our idea is inspired by AutoAugment [4], which exploited AutoML techniques [34] to automatically search for optimal augmentation policies to improve the model accuracy and generalization. However, it is difficult to apply this solution directly to our privacy problem. We need to address two new challenges: (1) how to efficiently evaluate the satisfaction of the two requirements for each policy (Sections 4.3 and 4.4); and (2) how to select the appropriate search space and sampling method (Section 4.4).

4.2. Privacy Score

During the search process, we need to quantify the privacy effect of the candidate policies. The PSNR metric is not efficient here, as it requires to perform an end-to-end reconstruction attack over a well-trained model. Instead, we design a new privacy score, which can accurately reflect the privacy leakage based on the transformation policy and a semi-trained model which is trained for only a few epochs.

We first define a metric `GradSim`, which measures the gradient similarity of two input samples (x_1, x_2) with the same label y:

$$\texttt{GradSim}(x_1, x_2) = \frac{\langle \bigtriangledown W(x_1, y), \bigtriangledown W(x_2, y) \rangle}{|| \bigtriangledown W(x_1, y)|| \cdot || \bigtriangledown W(x_2, y)||}. \qquad (5)$$

Assume the transformed image is $\widehat{x}$, which the adversary tries to reconstruct. He starts from a random input $x' = x_0$, and updates x' iteratively using Equation 1 until $\bigtriangledown W(x', y)$ approaches $\bigtriangledown W(\widehat{x}, y)$. Figure 1 visualizes this process: the y-axis is the gradient similarity $\texttt{GradSim}(x', \widehat{x})$, and x-axis is $i \in [0, 1]$ such that $x' = (1 - i) * x_0 + i * \widehat{x}$. The optimization starts with $i = 0$ (i.e., $x = x_0$) and ideally completes at $i = 1$ (i.e., $x' = \widehat{x}$ and $\texttt{GradSim} = 1$).

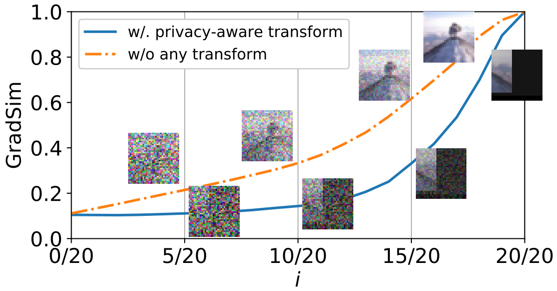


Figure 1: Visualization of the optimization process in reconstruction attacks.

A good policy can thwart the convergence from x_0 to $\widehat{x}$. As shown in Figure 1 (blue solid line), `GradSim` is hardly changed with i initially from x_0. This reveals the difficulty of the adversary to find the correct direction towards $\widehat{x}$ based on the gradient distance. In contrast, if the collaborative learning system does not employ any transformation function (red dash line), `GradSim` is increased stably with i. This gives the adversary an indication to discover the correct moving direction, and steadily make x' approach x by minimizing the gradient distance.

Based on this observation, we use the area under the `GradSim` curve to denote the effectiveness of a transformation policy in reducing privacy leakage. A good transformation policy will give a small area as the `GradSim` curve is flat for most values of i until there is a sharp jump when i is close to 1. In contrast, a leaky learning system has a larger area as the `GradSim` curve increases gradually with i. Formally, our privacy score is defined as below:

$$S_{pri}(T) = \frac{1}{|D|}\sum_{x\in D}\int_0^1 \texttt{GradSim}(x'(i), T(x))di,$$
$$x'(i) = (1-i)*x_0 + i*T(x). \tag{6}$$

For simplicity, we can approximate this score as a numeric integration, which is adopted in our implementation:

$$S_{pri}(T) \approx \frac{1}{|D|K}\sum_{x\in D}\sum_{j=0}^{K-1}\texttt{GradSim}(x'(\frac{j}{K}), T(x)). \tag{7}$$

Empirical validation. We also run some experiments to empirically verify the correlation between S_{pri} and PSNR. Specifically, we randomly select 100 transformation policies, and apply each to the training set. For each policy, we collect the PSNR value by performing the reconstruction attack [6] with a reduced iteration of 2500. We also measure the privacy score using Equation 7. As shown in Figure 2, S_{pri} is linearly correlated to PSNR with a Pearson Correlation Coefficient of 0.697. This shows that we can use S_{pri} to quantify the attack effects.

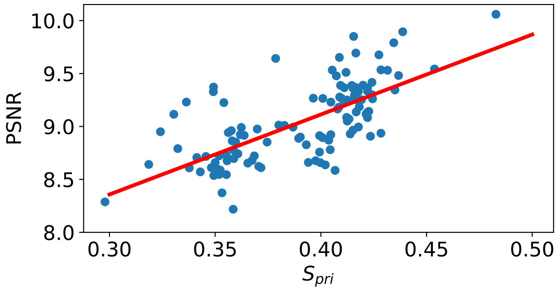


Figure 2: Correlation between PSNR and S_{pri}.

4.3. Accuracy Score

Another important requirement for a qualified policy is to maintain model accuracy. Certain transformations introduce large-scale perturbations to the samples, which can impair the model performance. We expect to have an efficient and accurate criterion to judge the performance impact of each transformation policy during the search process.

Joseph *et al.* [16] proposed a novel technique to search neural architectures without model training. It empirically evaluates the correlations between the local linear map and the architecture performance, and identifies the maps that yield the best performance. Inspired by this work, we adopt the technique to search for the transformations that can preserve the model performance.

Specifically, we prepare a randomly initialized model f, and a mini-batch of data samples transformed by the target policy T: $\{\widehat{x}_n\}_{n=1}^N$. We first calculate the Gradient Jacobian matrix, as shown below:

$$J = \begin{pmatrix}\frac{\partial f}{\partial \widehat{x}_1}, & \frac{\partial f}{\partial \widehat{x}_2}, & \cdots & \frac{\partial f}{\partial \widehat{x}_N}\end{pmatrix}^\top. \tag{8}$$

Then we compute its correlation matrix:

$$(M_J)_{i,j} = \frac{1}{N}\sum_{n=1}^{N} J_{i,n},$$
$$C_J = (J - M_J)(J - M_J)^T,$$
$$(\Sigma_J)_{i,j} = \frac{(C_J)_{i,j}}{\sqrt{(C_J)_{i,i}\cdot(C_J)_{j,j}}}. \tag{9}$$

Let $\sigma_{J,1} \leq \cdots \leq \sigma_{J,N}$ be the N eigenvalues of Σ_J. Then our accuracy score is given by

$$S_{acc}(T) = \frac{1}{N}\sum_{i=0}^{N-1} log(\sigma_{J,i}+\epsilon) + (\sigma_{J,i}+\epsilon)^{-1}, \tag{10}$$

where ϵ is set as 10^{-5} for numerical stability. This accuracy score can be used to quickly filter out policies which incur unacceptable performance penalty to the model.

4.4. Searching and Applying Transformations

We utilize the privacy and accuracy scores to identify the optimal policies, and apply them to collaborative training.

Search space. We consider the data augmentation library adopted by AutoAugment [4, 28]. This library contains 50 various image transformation functions, including rotation, shift, inversion, contrast, posterization, etc. We consider a policy combining at most k functions. This leads to a search space of $\sum_{i=1}^{k} 50^i$. Instead of iterating all the policies, we only select and evaluate C_{max} policies. For instance, in our implementation, we choose $k = 3$, and the search space is $127,550$. We set $C_{max} = 1,500$, which is large enough to identify qualified policies.

Search algorithm. Various AutoML methods have been designed to search for the optimal architecture, e.g., importance sampling [24], evolutionary sampling [19], reinforcement learning-based sampling [41]. We adopt a simple *random* search strategy, which is efficient and effective in discovering the optimal policies.

Algorithm 1 illustrates our search process. Specifically, we aim to identify a policy set $\mathcal{T}$ with n qualified policies. We need to prepare two local models: (1) M^s is used for privacy quantification. It is trained only with 10% of the original training set for 50 epochs. This overhead is equivalent to the training with the entire set for 5 epochs, which is very small. (2) M^r is a randomly initialized model without any optimization, which is used for accuracy quantification. We randomly sample C_{max} policies, and calculate the privacy and accuracy scores of each policy. The policies with accuracy scores lower than a threshold T_{acc} will be filtered out. We select the top-n policies based on the privacy score to form the final policy set $\mathcal{T}$.

Algorithm 1: Searching optimal transformations.

```
Input  : Augmentation library $\mathcal{P}$, $T_{acc}$, $C_{max}$, $M^s$, $M^r$, $D$
Output: Optimal policy set $\mathcal{T}$ with $n$ policies
1  for $i \in [1, C_{max}]$ do
2      Sample functions from $\mathcal{P}$ to form a policy $T$;
3      Calculate $S_{acc}(T)$ from $M^r$, $D$ (Eq. 10);
4      if $S_{acc}(T) \geq T_{acc}$ then
5          if $|\mathcal{T}| < n$ then
6              Insert $T$ to $\mathcal{T}$;
7          else
8              Calculate $S_{pri}(T)$ from $M^s$, $D$ (Eq. 7);
9              $T^* \leftarrow \mathrm{argmax}_{T' \in \mathcal{T}} S_{pri}(T')$;
10             if $S_{pri}(T) < S_{pri}(T^*)$ then
11                 Replace $T^*$ with $T$ in $\mathcal{T}$;
12 if $|\mathcal{T}| < n$ then
13     Go to Line 1;
14 return $\mathcal{T}$
```

Applying transformations. With the identified policy set $\mathcal{T}$, we can apply the functions over the sensitive training data. One possible solution is to always pick the policy with the smallest S_{pri}, and apply it to each sample. However, a single fixed policy can incur domain shifts and bias in the input samples. This can impair the model performance although we have tested it with the accuracy metric.

Instead, we can adopt a hybrid augmentation strategy which is also used in [4]: we randomly select a transformation policy from $\mathcal{T}$ to preprocess each data sample. All the selected transformation policies cannot have common transformation functions. This can guarantee low privacy leakage and high model accuracy. Besides, it can also improve the model generalization and eliminate domain shifts.

5. Experiments

5.1. Implementation and Configurations

Datasets and models. Our approach is applicable to various image datasets and classification models. Without loss of generality, we choose two datasets (CIFAR100 [18], Fashion MNIST [35]) and two conventional DNN models (ResNet20 [11], 8-layer ConvNet). These were the main targets of reconstruction attacks in prior works.

System and attack implementation. We implement a collaborative learning system with ten participants, where each one owns a same number of training samples from the same distribution. They adopt the SGD optimizer with momentum, weight decay and learning decay techniques to guarantee the convergence of the global model.

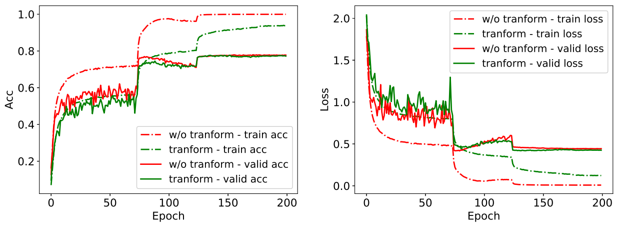
Our solution is able to thwart all existing reconstruction attacks with their variants. We evaluate six attacks in our experiments, named in the format of "optimizer+distance measure". These techniques[1] cover different optimizers and distance measures: (1) LBFGS+L2 [40]; (2) Adam+Cosine [6]; (3) LBFGS+Cosine; (4) Adam+L1; (5) Adam+L2; (6) SGD+Cosine. It is straightforward that the reconstruction attacks become harder with larger batch sizes. To fairly evaluate the defenses, we consider the strongest attacks where the batch size is 1.

Defense implementation. We adopt the data augmentation library [28], which contains 50 various transformations. We consider a policy with maximum 3 functions concatenated. It is denoted as $i - j - k$, where i, j, and k are the function indexes from [28]. Note that index values can be the same, indicating the same function is applied multiple times.

We implement the following defenses as the baseline.

- *Gaussian/Laplacian*: using differential privacy to obfuscate the gradients with Gaussian or Laplacian noise. For instance, Gaussian(10^{-3}) suggests a noise scale of $N(0, 10^{-3})$.
- *Pruning*: adopting the layer-wise pruning technique [2] to drop parameter gradients whose absolute values are

[1] The attack in [38] inherited the same technique from [40], with a smaller computational cost. So we do not consider it in our experiments.



(a) Accuracy

(b) Loss

Figure 3: Model performance of ResNet20 on CIFAR100 during the training process.

small. For instance, a compression ratio of 70% means for each layer, the gradients are set as zero if their absolute values rank after the top-30%.

- *Random augmentation*: we randomly sample transformation functions from [28] to form a policy. For each experiment, we apply 10 different random policies to obtain the average results.

We adopt PSNR to measure the visual similarity between the attacker's reconstructed samples and transformed samples, as the attack effects. We measure the trained model's accuracy over the corresponding validation dataset to denote the model performance.

Testbed configuration. We adopt PyTorch framework [26] to realize all the implementations. All our experiments are conducted on a server equipped with one NVIDIA Tesla V100 GPU and 2.2GHz Intel CPUs.

5.2. Search and Training Overhead

Search cost. For each transformation policy under evaluation, we calculate the average S_{pri} of 100 images randomly sampled from the validation set[2]. We also calculate S_{acc} with 10 forward-background rounds. We run 10 search jobs in parallel on one GPU. Each policy can be evaluated within 1 minutes. Evaluation of all $C_{max} = 1,500$ policies can be completed within 2.5 hours. The entire search overhead is very low. In contrast, the attack time of reconstructing 100 images using [6] is about 10 GPU hours.

Training cost. Applying the searched policies to the training samples can be conducted offline. So we focus on the online training performance. We train the ResNet20 model on CIFAR100 with 200 epochs. Figure 3 reports the accuracy and loss over the training and validation sets with and without our transformation policies. We can observe that although the transformation policies can slightly slow down the convergence speed on the training set, the speeds on the validation set are identical. This indicates the transformations incur negligible overhead to the training process.

5.3. Effectiveness of the Searched Policies

As an example, Figure 4 illustrates the visual comparison of the reconstructed images with and without the searched policies under the Adam+Cosine attack [6] for the two datasets. We observe that without any transformations, the adversary can recover the images with very high fidelity (row 2). In contrast, after the training samples are transformed (row 3), the adversary can hardly obtain any meaningful information from the recovered images (row 4). We have similar results for other attacks as well.

Table 1 reports the quantitative results of Adam+Cosine attacks and model accuracy. For each dataset and architecture, we consider the model training with no transformations, random selected policies, the top-2 of the searched policies and their hybrid. We observe that randomly selected policies fail to invalidate reconstruction attacks. In contrast, the searched policies can effectively reduce the deep leakage from the gradients. The hybrid of policies exhibits higher generalization ability on the final model.

Table 2 reports the PSNR values of the hybrid strategy against different reconstruction attacks and their variants. Compared with the training process without any defenses, the hybrid of searched transformations can significantly reduce the image quality of the reconstructed images, and eliminate information leakage in different attacks.

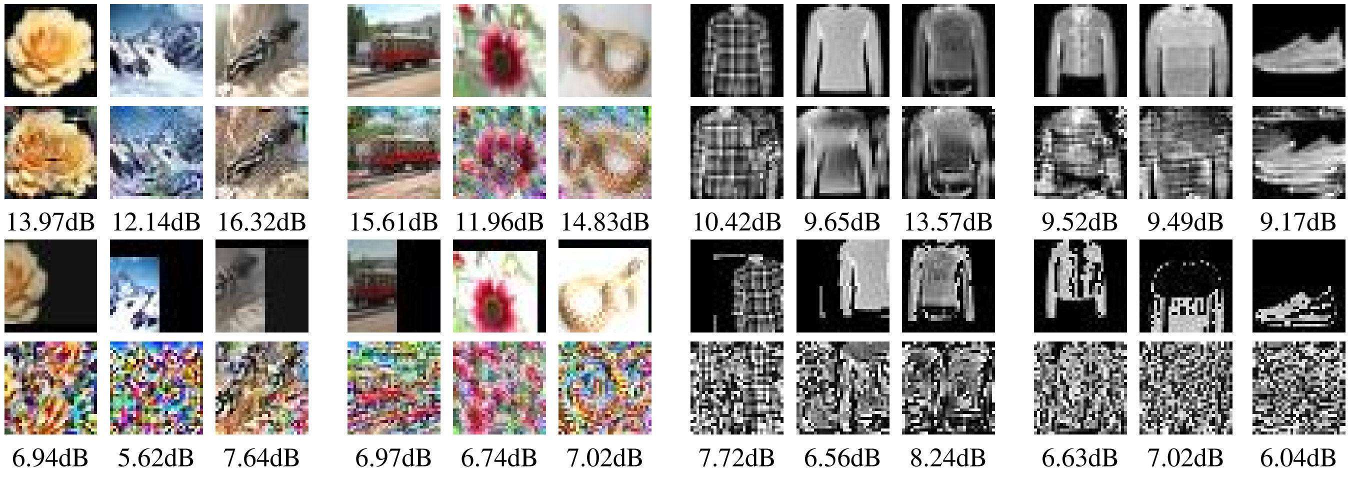
Comparisons with other defenses. We also compare our solution with state-of-the-art privacy-preserving methods proposed in prior works. We consider model pruning with different compression ratios, and differential privacy with different noise scales and types. Table 3 illustrates the comparison results. We observe that these solutions can hardly reduce the PSNR values, and the model accuracy is decreased significantly with larger perturbation. These results are consistent with the conclusion in [40]. In contrast, our solution can significantly destruct the quality of recovered images, while maintaining high model accuracy.

Transferability. In the above experiments, we search the optimal policies for each dataset. Actually the searched transformations have high transferability across different datasets. To verify this, we apply the policies searched from CIFAR100 to the tasks of F-MNIST, and Table 4 illustrates the PSNR and accuracy values. We observe that although these transferred policies are slightly worse than the ones directly searched from F-MNIST, they are still very effective in preserving the privacy and model performance, and much better than the randomly selected policies. This transferability property makes our solution more efficient.

5.4. Explanations about the Transformation Effects

In this section, we further analyze the mechanisms of the transformations that can invalidate reconstruction attacks. We first investigate which kinds of transformations are particularly effective in obfuscating input samples. Figure 5

[2] The first 100 images in the validation set are used for attack evaluation, not for S_{pri} calculation.



(a) CIFAR100 with ResNet20 (b) CIFAR100 with ConvNet (c) F-MNIST with ResNet20 (d) F-MNIST with ConvNet

Figure 4: Visual results and the PSNR values of the reconstruction attacks [6] with and without our defense. Row 1: clean samples. Row 2: reconstructed samples without transformation. Row 3: transformed samples. Row 4: reconstructed samples with transformation. The adopted transformations are the corresponding *Hybrid* policies in Table 1.

Policy	**PSNR**	**ACC**
None	13.88	76.88
Random	11.41	73.94
3-1-7	6.579	70.56
43-18-18	8.56	77.27
Hybrid	7.64	77.92

(a) CIFAR100 with ResNet20

Policy	**PSNR**	**ACC**
None	13.07	70.13
Random	12.18	69.91
21-13-3	5.76	66.98
7-4-15	7.75	69.67
Hybrid	6.83	70.27

(b) CIFAR100 with ConvNet

Policy	**PSNR**	**ACC**
None	10.04	95.03
Random	9.23	91.16
19-15-45	7.01	91.33
2-43-21	7.75	89.41
Hybrid	7.60	92.23

(c) F-MNIST with ResNet20

Policy	**PSNR**	**ACC**
None	9.12	94.25
Random	8.83	90.18
42-28-42	7.01	91.33
14-48-48	6.75	90.56
Hybrid	6.94	91.35

(d) F-MNIST with ConvNet

Table 1: PSNR (db) and model accuracy (%) of different transformation configurations for each architecture and dataset.

Attack	**None**	**Hybrid**	**Attack**	**None**	**Hybrid**
LBFGS+L2	6.93	4.79	LBFGS+COS	10.33	6.16
Adam+Cosine	13.88	7.64	Adam+L2	10.55	7.61
Adam+L1	9.99	6.97	SGD+COS	14.04	7.71

Table 2: The PSNR values (db) between the reconstructed and transformed images under different attack techniques.

Defense	**PSNR**	**ACC**
Pruning (70%)	12.00	77.12
Pruning (95%)	10.07	70.12
Pruning (99%)	10.93	58.33
Laplacian (10^{-3})	11.85	74.12
Laplacian (10^{-2})	9.67	39.59
Gaussian (10^{-3})	12.71	75.67
Gaussian (10^{-2})	11.44	48.2
Hybrid	7.64	77.92

Table 3: Comparisons with existing defense methods under the Adam+Cosine attack.

shows the privacy score of each transformation. The five transformations with the lowest scores are (red bars in the figure): 3rd [horizontal shifting, 9], 15th [brightness, 9], 18th [contrast, 7], 26th [brightness, 6] and 1st [contrast, 6]; where the parameters inside the brackets are the magnitudes of the transformations. These functions are commonly selected in the optimal policies.

Horizontal shifting achieves the lowest score, as it incurs a portion of black area, which can undermine the quality of the recovered image during the optimization. Contrast and brightness aim to modify the lightness of an image. These operations can blur the local details, which also increase the difficulty of image reconstruction. Overall, the selected privacy-preserving transformations can distort the details of the images, while maintaining the semantic information.

Next, we explore the attack effects at different network layers. We compare three strategies: (1) no transformation; (2) random transformation policy; (3) searched transformation policy. Figure 6 demonstrates the similarity between

Policy	PSNR	ACC
None	10.04	95.03
3-1-7	7.5	87.95
43-18-18	8.13	91.29
Hybrid	8.14	91.49

(a) F-MNIST with ResNet20

Policy	PSNR	ACC
None	9.12	94.25
21-13-3	7.51	74.81
7-4-15	7.68	88.29
Hybrid	7.11	87.51

(b) F-MNIST with ConvNet

Table 4: Transferability results: applying the same policies from CIFAR100 to F-MNIST.

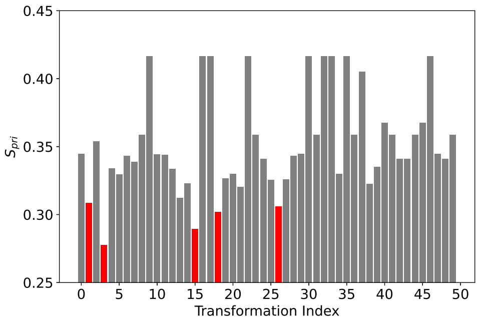
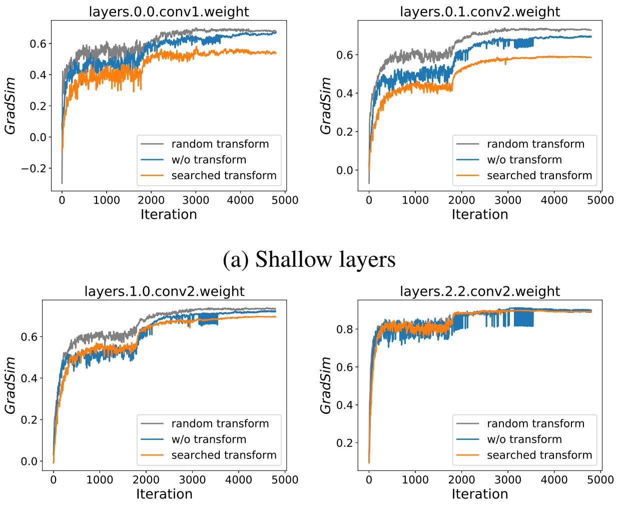


Figure 5: Privacy scores of the 50 transformation functions in the augmentation library.



(a) Shallow layers

(b) Deep layers

Figure 6: Gradient similarity during the reconstruction optimization process, for CIFAR100 with ResNet20.

the gradient of the reconstructed samples and the actual gradient for two shallow layers (a) and two deep layers (b). We can observe that at shallow layers, the similarity scores converge to 0.7 when no or random policy is applied. In contrast, the similarity score stays at lower values when the optimal policy is used. This indicates that the optimal policy makes it difficult to reconstruct low-level visual features of the input, e.g. color, shape, and texture. The similarity scores for all the three cases are almost the same at deep layers. This reveals the optimal policy has negligible impact on the semantic information of the images used for classification, and the model performance is thus maintained.

6. Discussions and Future Work

Adaptive attack. Our solution prevents image reconstruction via data augmentation techniques. Although the evaluations show it is effective against existing attacks, a more sophisticated adversary may try to bypass our defense from two aspects. First, instead of starting from a randomly initialized image, he may guess the content property or class representatives of the target sample, and start the reconstruction from an image with certain semantic information. The success of such attacks depends on the probability of a successful guess, which becomes lower with higher complexity or variety of images. Second, the adversary may design attack techniques instead of optimizing the distance between the real and dummy gradients. We leave these advanced attacks as future work.

Defending other domains. In this paper, we focus on the computer vision domain and image classification tasks. The reconstruction attacks may occur in other domains, e.g., natural language processing [40]. Then the searched image transformations cannot be applied. However, it is possible to use text augmentation techniques [17, 32] (e.g., deletion, insertion, shuffling, synonym replacement) to preprocess the sensitive text to be less leaky without losing the semantics. Future work will focus on the design of an automatic search method for privacy protection of NLP tasks.

7. Conclusion

In this paper, we devise a novel methodology to automatically and efficiently search for data augmentation policies, which can prevent information leakage from the shared gradients. Our extensive evaluations demonstrate that the identified policies can defeat existing reconstruction attacks with negligible overhead. These policies also enjoy high transferability across different datasets, and applicability to different learning systems. We expect our search method can be adopted by researchers and practitioners to identify more effective policies when new data augmentation techniques are designed in the future.

8. Acknowledgement

We thank the anonymous reviewers for their valuable comments. This research was conducted in collaboration with SenseTime. This work is supported by A*STAR through the Industry Alignment Fund — Industry Collaboration Projects Grant. It is also supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-019).

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC conference on Computer and Communications Security*, 2016. 2

[2] Dutta Aritra, Houcine Bergou El, M. Abdelmoniem Ahmed, Ho Chen-Yu, Narayan Sahu Atal, Canini Marco, and Kalnis Panos. On the discrepancy between the theoretical analysis and practical implementations of compressed communication for distributed deep learning. In *AAAI*, 2019. 5

[3] Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International Journal of Medical Informatics*, 2018. 1

[4] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Int. Conf. Comput. Vis.*, 2019. 3, 5

[5] Lixin Fan, Kam Woh Ng, Ce Ju, Tianyu Zhang, Chang Liu, Chee Seng Chan, and Qiang Yang. Rethinking privacy preserving deep learning: How to evaluate and thwart privacy attacks. *arXiv preprint arXiv:2006.11601*, 2020. 2

[6] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients–how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 3, 4, 5, 6, 7

[7] Shangwei Guo, Tianwei Zhang, Han Qiu, Yi Zeng, Tao Xiang, and Yang Liu. The hidden vulnerability of watermarking for deep neural networks. *arXiv preprint arXiv:2009.08697*, 2020. 3

[8] Shangwei Guo, Tianwei Zhang, Tao Xiang, and Yang Liu. Differentially private decentralized learning. *arXiv preprint arXiv:2006.07817*, 2020. 2

[9] Shangwei Guo, Tianwei Zhang, Xiaofei Xie, Lei Ma, Tao Xiang, and Yang Liu. Towards byzantine-resilient learning in decentralized systems. *arXiv preprint arXiv:2002.08569*, 2020. 1

[10] Meng Hao, Hongwei Li, Xizhao Luo, Guowen Xu, Haomiao Yang, and Sen Liu. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics*, 16(10), 2019. 1

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 5

[12] Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *ACM Annual Computer Security Applications Conference*, 2019. 2

[13] Zecheng He, Tianwei Zhang, and Ruby B Lee. Attacking and protecting data privacy in edge-cloud collaborative inference systems. *IEEE Internet of Things Journal*, 2020. 2

[14] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the GAN: information leakage from collaborative deep learning. In *ACM SIGSAC Conference on Computer and Communications Security*, 2017. 1

[15] Alain Hore and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. In *International Conference on Pattern Recognition*, 2010. 3

[16] Mellor Joseph, Turner Jack, Storkey Amos, and J. Crowley Elliot. Neural architecture search without training. *arXiv preprint arXiv:2006.04647*, 2020. 4

[17] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*, 2018. 8

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[19] Ngai Ming Kwok, Gu Fang, and Weizhen Zhou. Evolutionary particle filter: Re-sampling from the genetic algorithm perspective. In *International Conference on Intelligent Robots and Systems*, 2005. 5

[20] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *EEE Symposium on Security and Privacy*, 2019. 2

[21] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3), 1989. 2

[22] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE Symposium on Security and Privacy*, 2019. 1

[23] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy*, 2019. 1

[24] Radford M Neal. Annealed importance sampling. *Statistics and Computing*, 2001. 5

[25] Solmaz Niknam, Harpreet S Dhillon, and Jeffrey H Reed. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine*, 58(6), 2020. 1

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019. 6

[27] Hai Phan, My T Thai, Han Hu, Ruoming Jin, Tong Sun, and Dejing Dou. Scalable differential privacy with certified robustness in adversarial learning. In *International Conference on Machine Learning*, 2020. 2

[28] Philip Popien. AutoAugment - Learning Augmentation Policies from Data. `https://github.com/DeepVoltaire/AutoAugment`. 5, 6

[29] Han Qiu, Yi Zeng, Tianwei Zhang, Yong Jiang, and Meikang Qiu. Fencebox: A platform for defeating adversarial examples with data augmentation techniques. *arXiv preprint arXiv:2012.01701*, 2020. 3

[30] Han Qiu, Yi Zeng, Qinkai Zheng, Tianwei Zhang, Meikang Qiu, and Gerard Memmi. Mitigating advanced adversarial attacks with more advanced gradient obfuscation techniques. *arXiv preprint arXiv:2005.13712*, 2020. 3

[31] Han Qiu, Qinkai Zheng, Tianwei Zhang, Meikang Qiu, Gerard Memmi, and Jialiang Lu. Towards secure and efficient

deep learning inference in dependable iot systems. *IEEE Internet of Things Journal*, 2020. 3

[32] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019. 8

[33] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating gradient leakage attacks in federated learning. *arXiv preprint arXiv:2004.10397*, 2020. 1, 2

[34] Martin Wistuba, Ambrish Rawat, and Tejaswini Pedapati. A survey on neural architecture search. *arXiv preprint arXiv:1905.01392*, 2019. 3

[35] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[36] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2), 2019. 1

[37] Yi Zeng, Han Qiu, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. In *arXiv preprint arXiv:2012.07006*, 2021. 3

[38] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. iDLG: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020. 1, 2, 3, 5

[39] Qi Zhao, Chuan Zhao, Shujie Cui, Shan Jing, and Zhenxiang Chen. PrivateDL: Privacy-preserving collaborative deep learning against leakage from gradient sharing. *International Journal of Intelligent Systems*, 2020. 2

[40] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, 2019. 1, 2, 3, 5, 6, 8

[41] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016. 5