Automatic Transformation Search Against Deep Leakage from Gradients

Wei Gao, Xu Zhang, Shangwei Guo, Tianwei Zhang, *IEEE Member*, Tao Xiang, *IEEE Member*, Han Qiu, Yonggang Wen, *IEEE Fellow*, Yang Liu, *IEEE Senior Member*,

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 systematically analyze existing reconstruction attacks and propose to leverage data augmentation to defeat these attacks: by preprocessing sensitive images with carefully-selected transformation policies, it becomes infeasible for the adversary to extract training samples from the corresponding gradients. We first design two new metrics to quantify the impacts of transformations on data privacy and model usability. With the two metrics, we design a novel search method to automatically discover qualified policies from a given data augmentation library. Our defense method can be further combined with existing collaborative training systems without modifying the training protocols. We conduct comprehensive experiments on various system settings. Evaluation results demonstrate that the policies discovered by our method can defeat state-of-the-art reconstruction attacks in collaborative learning, with high efficiency and negligible impact on the model performance.

Index Terms—Collaborative learning, reconstruction attack, privacy protection, auto augmentation, transformation

1 Introduction

A collaborative learning system enables multiple participants to jointly train a shared Deep Learning (DL) model for a common artificial intelligence task [1]–[5]. Typical collaborative learning 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 (e.g., medical records, personally identifiable information) during the training phase. Due to these advantages, collaborative learning has become promising in many scenarios, e.g., smart manufacturing [6], autonomous driving [7], digital health [8], etc.

Although each participant does not disclose the training dataset, he has to share with others the gradients, from which an honest-but-curious adversary can inverse privacy information of the sensitive data indirectly. Past works [2], [9], [10] have demonstrated the possibility of membership inference and property inference attacks in collaborative learning. A more serious threat is the *reconstruction attack* [11]–[13], where an adversary can recover the exact values of samples from the shared gradients with high fidelity. In particular, reconstruction attacks start with "dummy" samples and labels and iteratively minimize the distance between the sharing gradients and the dummy gradients on the dummy points. This type of attacks 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 [11], [14] 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 has also been empirically evaluated [11]. 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*. The advantages of data augmentation are obvious. First, data augmentation has been widely applied as an effective methodology to improve generalization. Second, the transformations for data augmentation are lightweight and can be used without changing the protocol of the training process. 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 is challenging. First, privacy-enhanced data augmentation should ensure that the adversary cannot reconstruct the transformed sam-

- *Wei Gao and Xu Zhang contribute equally.*
- *Shangwei Guo is the corresponding author.*
- *Wei Gao, Tianwei Zhang, Yonggang Wen, and Yang Liu, are with School of Computer Science and Engineering, Nanyang Technological University, Singapore (Email: {gaow007, tianwei.zhang, ygwen, yangliu}@ntu.edu.sg).*
- *Xu Zhang, Shangwei Guo, and Tao Xiang are with College of Computer Science, Chongqing University, China (Email: {xuzhang, swguo, txiang}@cqu.edu.cn).*
- *Han Qiu is with the Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University and Zhongguancun Laboratory, Beijing, China. (e-mail: qiuhan@tsinghua.edu.cn).*

ples from the sharing gradients, and how to evaluate the privacy property of a given transformation is essential for designing such solution. But 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.

This paper extends our earlier work [15] and introduces a systematic approach to overcoming 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 modules to quantify the policies without training a complete model. First, we propose a novel privacy score to estimate the privacy leakage of gradients after using the corresponding transformation policy. Compared with our previous work, we optimize the privacy-preserving estimation by considering the performance of transformation policies over different classes and samples, i.e., the performance variance. We also theoretically and empirically show the effectiveness of our privacy score. Second, we adopt a training-free metric to filter policies that would be harmful to model performance. These metrics with our new search algorithm can identify the optimal policies within a few GPU hours from a data augmentation library consisting of 50 types of image transformation functions.

We conduct comprehensive experiments on different system settings to evaluate the performance of our method. The results show that the identified transformation policies exhibit great capability of preserving privacy against state-of-the-art reconstruction attacks 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 under various system settings. (2) The input transformations are performed locally without modifying the training pipeline. (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. The code is available at https://github.com/gaow0007/ATSPrivacy.

The contributions of this paper have four aspects.

- We systematically analyze and model reconstruction attacks and formalize the defense goals.
- We design and adopt two metrics to accurately quantify transformation policies on both privacy-preserving and model performance.
- We propose Automatic Transformation Search, a novel defense method to mitigate reconstruction attacks in collaborative learning systems. It searches for optimal policies from an augmentation library, which maintains the model performance and prevents deep leakage of gradients.
- We implement state-of-the-art reconstruction attacks and defense baselines and extensively evaluate our method on various system settings.

The rest of this paper is organized as follows. First, we provide background knowledge and introduce state-of-the-art techniques about defending against reconstruction attacks in Section 2. In Section 3, we formalize the definitions of the system and attacks and briefly describe the idea of our approach. In Section 4 and 5, we elaborated the design of our automatic augmentation search and show its superior privacy-preserving capability compared with existing baselines. Finally, the conclusion and future research directions are provided in Section 6.

2 Background and Related Work

2.1 Collaborative Learning

Collaborative learning enables multiple participants to collaboratively train a shared DL model [16]–[18]. Each participant updates the DL model on his privacy dataset using optimization methods such as stochastic gradient descent (SGD) and iteratively exchanges model updates without sharing his training samples. According to different communication network structures and parameter distribution, collaborative learning systems can be classified into two types: centralized and decentralized [19]–[21]. In the centralized architecture, one or multiple parameter servers are employed to coordinate the training process, including scheduling the training process, aggregating model updates, and distributing the aggregated updates and models. In the decentralized mode, each participant exchanges his updates with his neighbors and aggregates the received updates by himself using aggregation rules such as average aggregation, which mitigates the communication bottleneck in large collaborative learning systems. In each iteration of the training process, a participant first pulls the gradient from the parameter servers or neighbors, updates the gradient based on its local training samples, and shares the new gradient with the parameter servers or neighbors.

2.2 Auto Augmentation

Augmentation has been widely applied to train models in computer vision tasks, i.e. image classification, object detection, and video classification. During the training process, training images are preprocessed through a sequence of transformations. With the transformed images and their original labels, augmentation techniques can enhance the training dataset and improve model performance in the validation dataset. However, manual augmentation needs strong background knowledge and experience. Inspired by this problem, many search-based augmentation techniques have been proposed to automatically learn optimal augmentation policies from specific training datasets. The autoaugment techniques automatically search for improved data augmentation policies and the automatically selected augmentation policy contributes to the training dynamics of neural networks, e.g. generalization power. AutoAugment [22] usually contains two stages: 1) augmentation policy sampling, and 2) policy quality evaluation. For example, Cubuk et al. [22] present Autoaugment, which searches a policy with five sub-optimal transformation pairs and uses the validation accuracy in a hold-out dataset to update the policy prediction controller. To reduce the search overhead in Autoaugment, Lim et al. [23] proposed Fast Autoaugment, which splits the training dataset into multiple folds, each of which includes two subsets for training and policy evaluation. Instead of applying a global transformation policy over the whole dataset, Li et al. [24] develop a policy model to seek a sequence of image transformations for a single image.

Although AutoAugment has been widely adopted in AutoML fields [25] and makes impressive achievements across several computer vision tasks [26], [27], existing AutoAugment techniques focus on improving the accuracy performance on given tasks. In this paper, the AutoAugment philosophy is leveraged from the privacy-preserving perspective.

2.3 Reconstruction Attacks

Reconstruction attacks, also known as sample inference attacks, are privacy attacks where an adversary $\mathcal{A}$ tries to reconstruct the specific training samples from a given gradient of a participant. For example, in the image classification scenario, an adversary can reconstruct the training images and their labels of the current iteration from the shared gradient. Since collaborative learning systems adopt the gradient sharing framework, reconstruction attacks can be directly applied and have been becoming a pressing security problem in collaborative learning systems.

Zhu et al. [11] first formulate this attack as an optimization process: an adversarial participant uses L-BFGS [28] and search for the optimal samples whose corresponding gradients are close to the original ones. Following this work, several improved attacks were further proposed to enhance the attack performance and reduce the computational cost of the reconstruction. For instance, Zhao et al. [12] extract the training labels from the gradients and then recover the training samples with a higher convergence speed. Geiping et al. [13] adopt the cosine similarity as the distance function and Adam as the optimizer to solve the optimization problem, which can yield more precise reconstruction results. Yin et al. [29] proposed large batch image reconstruction from averaged gradients, which utilizes Batch-Normalization to acquire better image reconstruction results. Jeon et al. [30] and Li et al. [31] combine a gradient matching process with GANs to reconstruct images in the determined distribution. Hatamizadeh et al. [32] adapt existing attacks to transformer-based models by introducing additional constraints between image patches. Zhang et al. [33] perform reconstruction attacks against various transformer architectures and demonstrate that the transformer-based models are more susceptible to recent reconstruction attacks.

2.4 Defenses

Gradient modification To mitigate reconstruction attacks in collaborative learning, one straightforward strategy is to obfuscate the gradients before releasing them, in order to make the reconstruction difficult or infeasible. For instance, differential privacy is a theoretical framework to guide the randomization of exchange information [34]–[37]. Zhu et al. [11] tried to add Gaussian/Laplacian noises guided by differential privacy to the gradients and compress the model with gradient pruning to mitigate reconstruction attacks. 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 a certain amount of information. Besides, this approach cannot defeat the reconstruction attacks and fails to protect the training points even when the perturbations and the pruning ratio are very large. Such ineffectiveness of these methods was validated in [11], and will be further confirmed in this paper (Table 4). Wei et al. [14] proposed to adjust the hyperparameters (e.g. bath size, loss, or distance function), which also has a limited impact on the attack results.

Architectures modification An alternative direction is to design new collaborative learning systems to thwart the reconstruction attacks. Zhao et al. [38] 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. Bonawitz et al. [39] presented a secret aggregation framework for collaborative learning through sending secret gradients based on homomorphic encryption. Fan et al. [40] designed a secret polarization network for each participant to produce secret losses and calculate the gradients. Sun et al. [41] insert a defense layer following fully connected layers to prune features that are important for the reconstruction of the adversary. This approach requires all participants to follow new training pipelines or optimization methods. They cannot be directly applied to existing collaborative implementations. This significantly restricts their practicality.

Dataset condensation. Alternatively, the data condensation strategy [42]–[47] is proposed to achieve efficient or private-preserving model training. Some methods [42], [43] generate samples that are similar to the original distribution via generative models (e.g., GANs). Others [44], [45], [47] utilize optimization techniques to mitigate the gap between the synthetic datasets and their target ones. However, these solutions require all participants to train powerful GANs and consume massive computational resources and data. Otherwise, the synthetic samples will seriously affect the accuracy of the shared model. This is very costly for collaborative learning and becomes difficult for resource-constrained participants. We perform experiments to comprehensively compare these methods with ours in Section 5.3.

3 Problem Statement

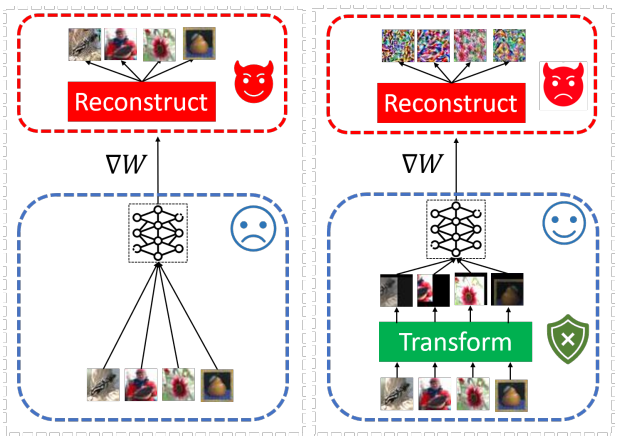
3.1 System Model

We consider a standard collaborative learning system where all participants jointly train a global model M. Each participant owns a private dataset D from an unknown distribution. 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 gradients are shared with other participants or parameter servers for aggregation. The participants can also use the mini-batch SGD, where a mini-batch of samples is randomly selected to generate the gradient at each iteration.

3.2 Attack Model

We consider an 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. Fig. 1a illustrates the reconstruction attacks in collaborative learning systems.

Common reconstruction techniques [11]–[13] 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



(a) (b)

Fig. 1: Reconstruction attacks (a) and our proposed approach (b) in collaborative learning systems.

can be formulated as an optimization problem of minimizing the objective:

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

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) [48]. 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:*

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

3.3 Our Approach

Driven by the severity of reconstruction attacks and the limitations of existing defenses, we focus on a new mitigation opportunity: *transforming the sensitive training samples to make the reconstruction difficult or even infeasible*. Image transformation has been widely adopted to mitigate adversarial examples [49]–[51], backdoor attacks [52], and attack watermarking schemes [53]. We repurpose it for defeating reconstruction attacks in collaborative learning systems. 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) Privacy-preserving: with the transformation policy, the adversary fails to reconstruct the corresponding training points from the perturbed gradient at the current iteration, i.e., the adversarial participant is not able to infer $\widehat{x}$ (and x) from $\bigtriangledown W(\widehat{x}, y)$ with the diverse label y.
2) Performance-preserving: the final global model should maintain a similar performance as the one trained from the datasets without the transformation policy.

In the context of collaborative learning, each participant can identify the optimal transformations from his/her dataset, and preprocess the private samples before local training. 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 $(\epsilon, \delta, \gamma)$-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 $\texttt{ACC}$ is the prediction accuracy function.

It is critical to identify the transformation functions that can satisfy the above two requirements, i.e., transformation functions with lower ϵ, γ. 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.

4 Methodology

4.1 Overview

In this section, we introduce a systematic and automatic method to search for privacy-preserving and efficient policies from a large quantity and variety of augmentation functions, which can prevent reconstruction attacks while still providing acceptable model performance. Our idea is inspired by AutoAugment [22] that exploits AutoML techniques [25] 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; and (2) how to select the appropriate search space and sample method.

For the first challenge, one intuitive way is to randomly search transformation policies and quantitatively evaluate their performance in defending against reconstruction attacks using existing image quality evaluation metrics e.g. PSNR and SSIM. However, such an approach requires performing an end-to-end reconstruction attack over a well-trained model. It is infeasible to train a model and then measure the model performance and privacy leakage. Thus, 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. For performance estimation, existing NAS techniques have shown that training-free metrics could efficiently estimate the model performance on certain datasets. We slightly modify one of them as our accuracy score to filter out policies that have a seriously negative impact on model performance.

For the second challenge, to trade off effectiveness and preprocessing overhead, we first adopt one augmentation library that involves common transformation policies. Then, we design a novel search strategy to find optimal transformation policies from the library based on the proposed privacy and accuracy scores. We will demonstrate that our augmentation library is effective enough for defeating reconstruction attacks in the experiments. Finally,

we apply our strategies to overcome the challenges in collaborative learning systems, integrated with new privacy (Section 4.2) and performance (Section 4.3) metrics, and searching and training protocols (Section 4.4). Fig. 2 illustrates the pipeline to identify optimal transformation policies. Below we describe each technique in detail.

4.2 Privacy Score

During the search process, we need to quantify the privacy effect of the candidate policies. The straightforward way to quantify the privacy effect of a policy is to evaluate the image quality of reconstruction attacks over a will-trained model. However, such image quality assessments (e.g., PSNR) are not efficient here, as it requires performing an end-to-end reconstruction attack over a well-trained model. On the other hand, as shown in Section 3.2, the adversary adopts optimization techniques to recover the target images by gradually reducing the distance between gradients. Thus, policies with satisfactory privacy protection ability should thwart the optimization process and make the training images difficult to be converged. Following this analysis, we propose a privacy score that 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 when given the same model parameters W:

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

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 $\bigtriangledown W(\widehat{x}, y)$. Fig. 3 visualizes the GradSim scores across different interpolation coefficients: 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 $\texttt{GradSim} = 1$).

AUC. A good policy can thwart the convergence from x_0 to $\widehat{x}$. As shown in Fig. 3 (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 employs the standard transformation (yellow dash line), e.g., random crop, GradSim is increased stably with i. This gives the adversary an indication to discover the correct moving direction and steadily makes x' approach x by minimizing the gradient distance.

Based on this observation, we use the Area Under the GradSim Curve (AUC), $\mathcal{S}$, to denote the effectiveness of a transformation policy T in reducing privacy leakage on image x, i.e.,

$$\mathcal{S}(x, \widehat{x}) = \int_0^1 \texttt{GradSim}(x'(i), \widehat{x}) di. \quad (6)$$

A good transformation policy will give a small $\mathcal{S}$ 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 $\mathcal{S}$ as the GradSim curve increases gradually with i.

Theoretical analysis. Our proposed AUC simulates the advantage of the attacker during the reconstruction optimization under the protection of a transformation policy and is correlated to the image quality of the reconstructed input samples when the *GradSim* values satisfy some given conditions. This observation can be verified by the following theorem.

Theorem 1. *Consider two transformation policies T_1, T_2. $\widehat{x}_1 = T_1(x)$ and $\widehat{x}_2 = T_2(x)$ for a training input x. Let $x'_{1,k}, x'_{2,k}$ be the k-th reconstruction results of $\widehat{x}_1$ and $\widehat{x}_2$ via Equation 1. If $\texttt{GradSim}(x'_{1,k}, \widehat{x}_1) \geq \texttt{GradSim}(x'_{2,k}, \widehat{x}_2)$ and the GradSim increase of reconstructing $\widehat{x}_1$ from the k-th to the k+1-th iteration is larger than that of reconstructing $\widehat{x}_2$, then T_2 is more effective against reconstruction attacks than T_1 at the k-th iteration.*

Proof. Since $\textit{GradSim}(x'_{1,k}, \widehat{x}_1) \geq \textit{GradSim}(x'_{2,k}, \widehat{x}_2)$, we have

$$\begin{aligned} &\textit{GradSim}(x'_{1,k}, \widehat{x}_1) - \textit{GradSim}(x'_{2,k}, \widehat{x}_2) \\ &= \frac{\langle \bigtriangledown W(x'_{1,k}, y), \bigtriangledown W(\widehat{x}_1, y)\rangle}{|| \bigtriangledown W(x'_{1,k}, y)|| \cdot || \bigtriangledown W(\widehat{x}_1, y)||} \\ &\quad - \frac{\langle \bigtriangledown W(x'_{2,k}, y), \bigtriangledown W(\widehat{x}_2, y)\rangle}{|| \bigtriangledown W(x'_{2,k}, y)|| \cdot || \bigtriangledown W(\widehat{x}_2, y)||} \\ &\geq 0. \end{aligned} \quad (7)$$

Without loss of generality, we assume the two reconstruction processes use the same attack parameters and normalize the gradients. Then, $\langle \bigtriangledown W(x'_{1,k}, y), \bigtriangledown W(\widehat{x}_1, y)\rangle - \langle \bigtriangledown W(x'_{2,k}, y), \bigtriangledown W(\widehat{x}_2, y)\rangle \geq 0$. Because the *GradSim* increase of reconstructing $\widehat{x}_1$ from the k-th to the k+1-th iteration is larger than that of reconstructing $\widehat{x}_2$, then

$$\begin{aligned} &\textit{GradSim}(x'_{1,k+1}, \widehat{x}_1) - \textit{GradSim}(x'_{1,k}, \widehat{x}_1) \\ &\geq \textit{GradSim}(x'_{2,k+1}, \widehat{x}_2) - \textit{GradSim}(x'_{2,k}, \widehat{x}_2). \end{aligned}$$

Similarly, we have

$$\begin{aligned} &\langle \bigtriangledown W(x'_{1,k}, y), \bigtriangledown W(\widehat{x}_1, y)\rangle - \langle \bigtriangledown W(x'_{1,k+1}, y), \bigtriangledown W(\widehat{x}_1, y)\rangle \\ &\geq \langle \bigtriangledown W(x'_{2,k}, y), \bigtriangledown W(\widehat{x}_2, y)\rangle - \langle \bigtriangledown W(x'_{2,k+1}, y), \bigtriangledown W(\widehat{x}_2, y)\rangle. \end{aligned}$$

Then,

$$\begin{aligned} &\langle \bigtriangledown W(x'_{1,k}, y) - \bigtriangledown W(x'_{1,k+1}, y), \bigtriangledown W(\widehat{x}_1, y)\rangle \\ &\geq \langle \bigtriangledown W(x'_{2,k}, y) - \bigtriangledown W(x'_{2,k+1}, y), \bigtriangledown W(\widehat{x}_2, y)\rangle. \end{aligned} \quad (8)$$

Thus, the gradient difference between $x'_{1,k}$ and $x'_{1,k+1}$ is larger than that between $x'_{2,k}$ and $x'_{2,k+1}$. Therefore, according to the reconstruction analysis of [54] and Equations 7-8, for the k+1-th reconstruction iteration of $\widehat{x}_1$, the attacker can obtain $x'_{1,k+1}$ that is expected to be closer to $\widehat{x}_1$ than that of $\widehat{x}_2$. □

Variance. Due to the huge content diversity of images between different classes, the GradSim value of a transformation policy T varies on different classes. Besides, we can hardly evaluate the performance of a policy on all images sampled from the unknown distribution. To keep the consistency of different classes and guarantee the privacy protection on the entire dataset, we use another component: the GradSim variance among different classes and samples, $\texttt{Var}_T$, to evaluate the overall performance of a transformation policy T, where

$$\texttt{Var}_T = \frac{\Sigma_{x \in D}(\mathcal{S}(x, \widehat{x}) - \bar{\mathcal{S}})^2}{|D|}, \quad (9)$$

where $\bar{\mathcal{S}}$ is the average of the AUC scores on all samples of D. A smaller $\texttt{Var}_T$ value indicates the corresponding transformation

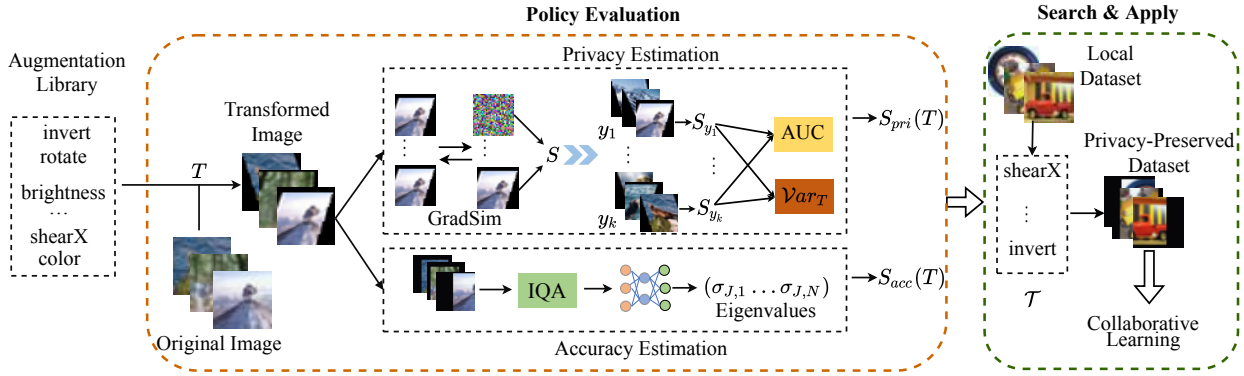


Fig. 2: Pipeline to identify optimal transformation policies using our privacy and accuracy scores.

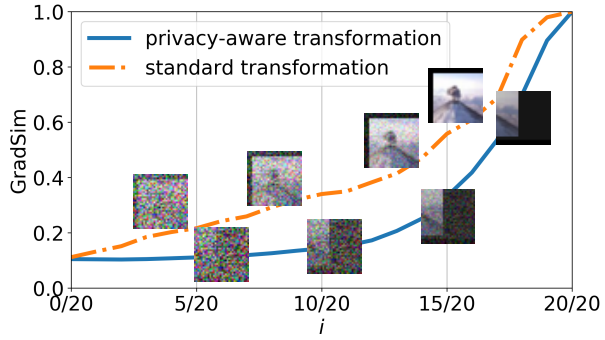


Fig. 3: Visualization of calculating S_{pri}.

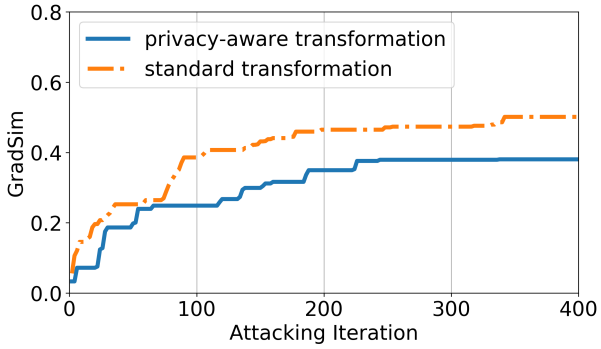


Fig. 4: The reconstruction processes of different transformations used in Fig. 3.

policy has consistent performance on images of different classes and samples when defeating reconstruction attacks. Note that D is expected to contain samples of all classes.

To obtain the final privacy score of a transformation policy, we adaptively combine the AUC and Variance components together. Formally, our privacy score is defined as below:

$$S_{pri}(T) = \frac{\alpha}{|D|} \sum_{x \in D} \mathcal{S}(x, \widehat{x}) + \beta \texttt{Var}_T. \tag{10}$$

where α and β are the weights of AUC and Var, respectively. $\alpha + \beta = 1$. For simplicity, we can approximate this score as a numeric integration, which is adopted in our implementation:

$$S_{pri}(T) \approx \frac{\alpha}{|D|K} \sum_{x \in D} \sum_{j=0}^{K-1} \texttt{GradSim}(x'(\frac{j}{K}), \widehat{x}) + \beta \texttt{Var}_T,$$
$$x'(i) = (1-i) * x_0 + i * \widehat{x}. \tag{11}$$

Empirical validation. In this study, we use the linear interpolation to approximate the reconstruction optimization procedure. However, it is worth noting that a gap exists between the linear interpolation and the actual reconstruction attack. This is demonstrated in Fig.4, where we observe a rapid increase in GradSim at the beginning of the reconstruction attack. Although the curve associated with the reconstruction attack differs from that in the linear interpolation, the privacy-aware transformation we select based on the privacy score can still effectively mitigate the threat of reconstruction attacks. Additionally, we 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

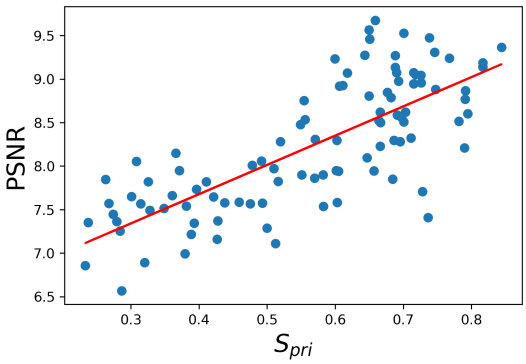


Fig. 5: Correlation between PSNR and S_{pri}.

PSNR values by performing the reconstruction attack [13] with a reduced iteration of 2500. We also measure the privacy score using Equation 11. As shown in Fig. 5, S_{pri} is linearly correlated to PSNR with a Pearson Correlation Coefficient of 0.755. This shows that we can use S_{pri} to quantify the attack effects.

4.3 Accuracy Score

Although several transformations enlarge the privacy scores of training images, they may also introduce large-scale perturbations to the samples, which can impair the model performance. Thus, another requirement for searching for qualified policies is to effectively estimate the model performance on the transformed dataset. In our methodology, we adopt an efficient and accurate score to evaluate the performance impact of each transformation policy during the search process.

Some training-free metrics have been proposed to estimate the model performance on the specific dataset [55]–[58]. Considering the computation cost and effectiveness, we apply the metric in [58] to quantize the performance impact of policies. It empirically evaluates the correlations between the local linear map and the architecture performance, and identifies the maps that yield the best performance. We adopt this technique to search for 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:

$$J = \left(\frac{\partial f}{\partial \widehat{x}_1}, \quad \frac{\partial f}{\partial \widehat{x}_2}, \quad \cdots \quad \frac{\partial f}{\partial \widehat{x}_N}\right)^{\top}. \tag{12}$$

Then we compute its correlation matrix:

$$\begin{aligned} (M_J)_{i,j} &= \frac{1}{N}\sum_{n=1}^{N} J_{i,n}, \\ C_J &= (J - M_J)(J - M_J)^T, \\ (\Sigma_J)_{i,j} &= \frac{(C_J)_{i,j}}{\sqrt{(C_J)_{i,i} \cdot (C_J)_{j,j}}}. \end{aligned} \tag{13}$$

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

$$S_{acc}(T) = \frac{1}{N}\sum_{i=0}^{N-1} log(\sigma_{J,i} + \eta) + (\sigma_{J,i} + \eta)^{-1}, \tag{14}$$

where η is set as 10^{-5} for numerical stability in our experiments. The calculation of $S_{acc}(T)$ is inspired by the previous work [58], which evaluates the correlations between the Jacobian correlation and the architecture performance. Similarly, we adopt the Jacobian correlations to identify transformation policies that can yield better model performance. Thus, This accuracy score can be used to quickly filter out policies that incur unacceptable performance penalties to the model. Note that the accuracy estimation is conducted on the transformed samples, while normal inputs would not be distorted during the inference process. Thus, the features learned by the collaboratively trained model may be different from those in neural real samples. In our experiments, before applying the accuracy score, we filter policies that would cause huge distortions using existing image quality assessments (IQA) such as FSIM [59] to speed up the performance evaluation process.

4.4 Searching Transformations

Search space. We consider the data augmentation library adopted by AutoAugment [22], [60]. This library contains 50 various image transformation functions, including rotation, shift, inversion, contrast, posterization, etc. We will describe each transformation function with the corresponding magnitude in the experiments. 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 [61], evolutionary sampling [62], reinforcement learning-based sampling [63]. 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. As discussed in [64], it is relatively easy to reconstruct inputs using the randomly initialized model. To enlarge the gap between the privacy scores of the good and poor augmentation policies, we prepare a local model M^s for privacy quantification, which 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. Besides, We follow the practice of [58] to set a randomly initialized model M^r for accuracy quantification. We randomly sample C_{max} policies, and calculate the privacy and accuracy scores of each policy. The policies with lower accuracy scores 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.

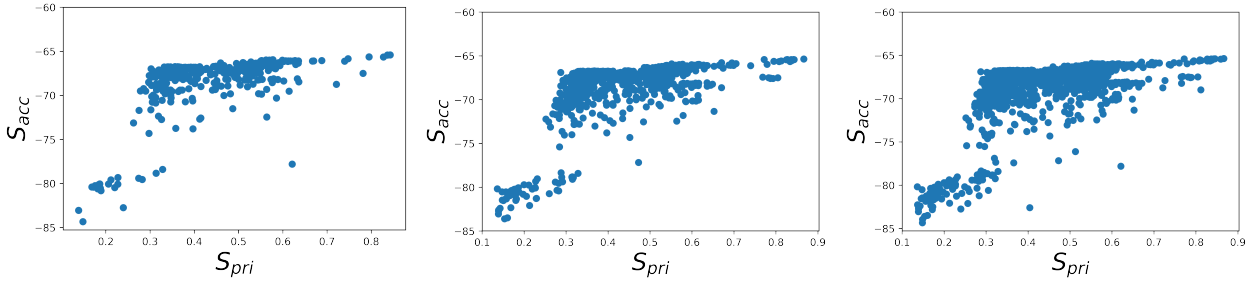
Input : Augmentation library $\mathcal{P}$, T_{acc}, C_{max}, M^s, D
Output: Optimal policy set $\mathcal{T}$ with n policies

```
for i ∈ [1, C_max] do
    Sample functions from P to form a policy T;
    Randomly initialize a model M^r;
    Calculate S_acc(T) from M^r, D (Eq. 14);
    if S_acc(T) ≥ T_acc then
        if |T| < n then
            Insert T to T;
        else
            Calculate S_pri(T) from M^s, D (Eq. 11);
            T* ← argmax_{T'∈T} S_pri(T');
            if S_pri(T) < S_pri(T*) then
                Replace T* with T in T;
if |T| < n then
    Go to Line 1;
return T
```

4.5 Privacy-preserving Collaborative Training

Hybrid augmentation. 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 metrics. Instead, we can adopt a hybrid augmentation strategy which is also used in [22]: 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.

Collaborative training. Algorithm 2 illustrates the privacy-preserving collaborative training process for each participant. Our methodology is to find optimal augmentation policies against reconstruction attacks. Thus, the training process is the same as the normal one, except the normal augmentation policies are replaced by the searched transformation policies. Since our defense method does not affect the original training process, it can be integrated with various collaborative learning systems.



(a) $S_{pri} \in [0.138, 0.843]$ (b) $S_{pri} \in [0.136, 0.866]$ (c) $S_{pri} \in [0.136, 0.866]$

Fig. 6: $S_{pri} - S_{acc}$ distributes of different numbers of policies. The numbers of policies for (a), (b), (c) are 500, 1500, and 5000 respectively. We observe that the privacy score range of 1500 policies is the same as that of 5000. Therefore, we set 1500 as the policy search space in our experiment.

Algorithm 2: Applying transformation policies to collaboratively train the shared model for each participant.

Input : Optimal set $\mathcal{T}$, D, iteration number N
for $i \in [1, N]$ **do**
  Sample a mini-batch B from D;
  Initialize $\widehat{B}$ as empty;
  for $x \in B$ **do**
    Randomly select a policy T from $\mathcal{T}$;
    $\widehat{x} = T(x)$;
    Insert $\widehat{x}$ to $\widehat{B}$;
  Compute the local gradient g_i on $\widehat{B}$;
  Share g_i with the parameter servers or neighbors;
  /* *The parameter servers or neighbors aggregate the received gradients and update the shared model* */
  Receive the aggregated model M_i;
return M_N

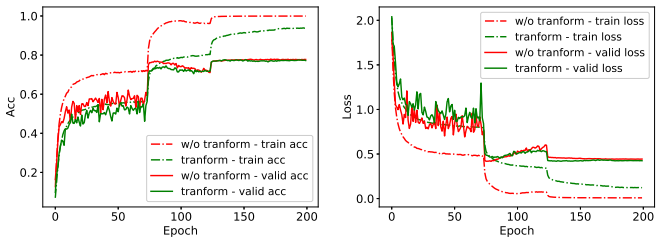
5 Experiments

In this section, we first introduce the experimental configurations and the implementations of attack and defense methods. Then, we present the experimental results of the searched policies and the corresponding analysis.

5.1 Configurations

Datasets and models. Our approach is applicable to various image datasets and classification models. Without loss of generality, we first choose two datasets (CIFAR100 [65], Fashion MNIST (F-MNIST) [66]) and three conventional DNN models (ResNet20 [67], 8-layer ConvNet for F-MNIST, 32-layer ConvNet for CIFAR100). These were the main targets of reconstruction attacks in prior works. To demonstrate the generalizability of our method, we also conduct comprehensive experiments on different model architectures (i.e., MobileNet [68], ViT [69]) and high-resolution image datasets (i.e., ImageNet [26], CelebA [70]).

Training configurations. We implement a collaborative learning system with ten participants, where each one owns the 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. In particular, we utilize SGD with momentum 0.9 and weight decay $5 \cdot 10^{-4}$ to optimize the deep neural networks and set the training epoch as 200 (resp. 100) for CIFAR100 (resp. F-MNIST). The initial learning rate is 0.1 and steply decays by a factor of 0.1 at $\frac{3}{8}$, $\frac{5}{8}$ and $\frac{7}{8}$ of all iterations. Besides, we also adopt typical default augmentation policies to train different models, e.g., RandomCrop for CIFAR100 and F-MNIST.



(a) Accuracy (b) Loss

Fig. 7: Model performance of ResNet20 on CIFAR100 during the training process.

Attack implementation. 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) DLG (LBFGS+L2) [11] that adopts LBFGS and L2 norm for optimization; (2) IG (Adam+Cosine) [13] that adopts Adam and cosine distance for optimization; (3) LBFGS+Cosine, a variant of DLG that replaces L2 norm with cosine distance; (4) Adam+L1, a variant of IG that replaces cosine distance with L1 norm; (5) GI (Adam+L2) [29] that adopts Adam and L2 norm for optimization; (6) SGD+Cosine, a variant of IG that replaces Adam with SGD. It is straightforward that the reconstruction attacks become harder with larger batch sizes. To fairly evaluate the defenses, we mainly consider the strongest attacks where the batch size is 1.

For (1)(2)(5), we follow the same setting in their corresponding papers [11], [13], [29]. Because the results of L-BFGS-based reconstruction attacks are unstable, we run the attacks 16 times and select the best reconstruction results for the attacks with L-BFGS optimizer (i.e., (1), (6)). For a fair comparison, we reduce the iteration number to 300. For the remaining attacks (4), (6), we apply the same configuration with (2).

Defense implementation. As shown in Table 1, we adopt the data augmentation library [60], which contains 50 various trans-

1. The attack in [12] inherited the same technique from [11], with a smaller computational cost. So we do not consider it in our experiments.

TABLE 1: Summary of the 50 Transformations

Index	Transformation	Magnitude	Index	Transformation	Magnitude	Index	Transformation	Magnitude	Index	Transformation	Magnitude
0	invert	7	13	brightness	5	26	brightness	6	39	color	0
1	contrast	6	14	sharpness	9	27	color	8	40	solarize	1
2	rotate	2	15	brightness	9	28	solarize	0	41	autocontrast	0
3	translateX	9	16	translateX	4	29	invert	0	42	translateY	3
4	sharpness	1	17	equalize	1	30	equalize	0	43	translateY	4
5	sharpness	3	18	contrast	7	31	autocontrast	0	44	autocontrast	1
6	shearY	2	19	sharpness	5	32	equalize	8	45	solarize	1
7	translateY	2	20	color	5	33	equalize	4	46	equalize	5
8	autocontrast	5	21	translateX	5	34	color	5	47	invert	1
9	equalize	2	22	equalize	7	35	equalize	5	48	translateY	3
10	shearY	5	23	autocontrast	8	36	autocontrast	4	49	autocontrast	1
11	posterize	5	24	translateY	3	37	solarize	4			
12	color	3	25	sharpness	6	38	brightness	3			

formations. We consider a policy with a maximum of 3 functions concatenated. It is denoted as $i - j - k$, where i, j, and k are the function indexes from [60]. Note that the index values can be the same, indicating the same function is applied multiple times. The search space is $\sum_{i=1}^{3} 50^i = 127,550$, which is redundant and impossible to implement the search algorithm. We test 500, 1500, and 5000 policies and plot the privacy-training accuracy distribution in Fig. 6. We observe our defenses with 1500 and 5000 candidates have similar performance and the privacy score ranges are almost the same, which indicates that 1500 policies are enough for searching for satisfactory transformation policies.

We implement the following defenses as the baseline.

- *Random augmentation*: we randomly sample transformation functions from [60] to form a policy. For each experiment, we apply 10 different random policies to obtain the average results.
- *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 [71] to drop parameter gradients whose absolute values are 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%.
- *InstaHide*: mixing multiple images and randomly flipping the sign of pixels to protect images. For instance, InstaHide(4) means that a composite image is merged with four images.

We also implement state-of-the-art dataset condensation methods [43]–[45], [47], [54] and compare our method with these baselines in the experiments.

Instead of searching transformation policies for each participant, we first adopt Algorithm 1 to obtain a universal policy set $\mathcal{T}$ on a validation set. During the training process, all participants would use $\mathcal{T}$ for privacy protection. In particular, for each policy T, we randomly sample 10 images from every label data to compute the corresponding privacy score $S_{pri}(T)$ and set $\alpha = 0.7$, $\beta = 0.3$. We optimize the randomly initialized model f for 10 forward-background rounds and use the average value of the training accuracy scores of the ten rounds as $S_{acc}(T)$.

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 [72] 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 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. For 100 images, each policy can be evaluated within 1 minute and 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 [13] 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. Fig. 7 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

Our defense method can search for satisfactory transformation policies that are more effective in defending against state-of-the-art reconstruction attacks than the involved baselines. Fig. 8 and 9 illustrate the visual comparison of the reconstructed images with and without the searched policies under the Adam+Cosine attack [13] for the CIFAR100 and F-MNIST datasets, respectively. We observe that without any transformations, the adversary can recover the images with very high fidelity (Row 2, CIFAR100 with ResNet20 and ConvNet, Fig. 8). 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 on F-MNIST (Fig. 9) and for other attacks.

Table 2 reports the quantitative results of the Adam+Cosine attack and model accuracy. For each dataset and architecture, we consider the model training with standard transformations, randomly selected policies, hybrid policies chosen by our earlier work [15], the top-2 of our searched policies and their hybrid version. Note that we use standard augmentation policies for model training, but do not apply any augmentation for reconstruction attacks, following the common practice adopted by previous works [11]–[13].

We observe that the randomly selected policies fail to invalidate reconstruction attacks. In contrast, our searched policies can effectively reduce the deep leakage from the gradients and the hybrid of policies exhibits higher generalization ability on the

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

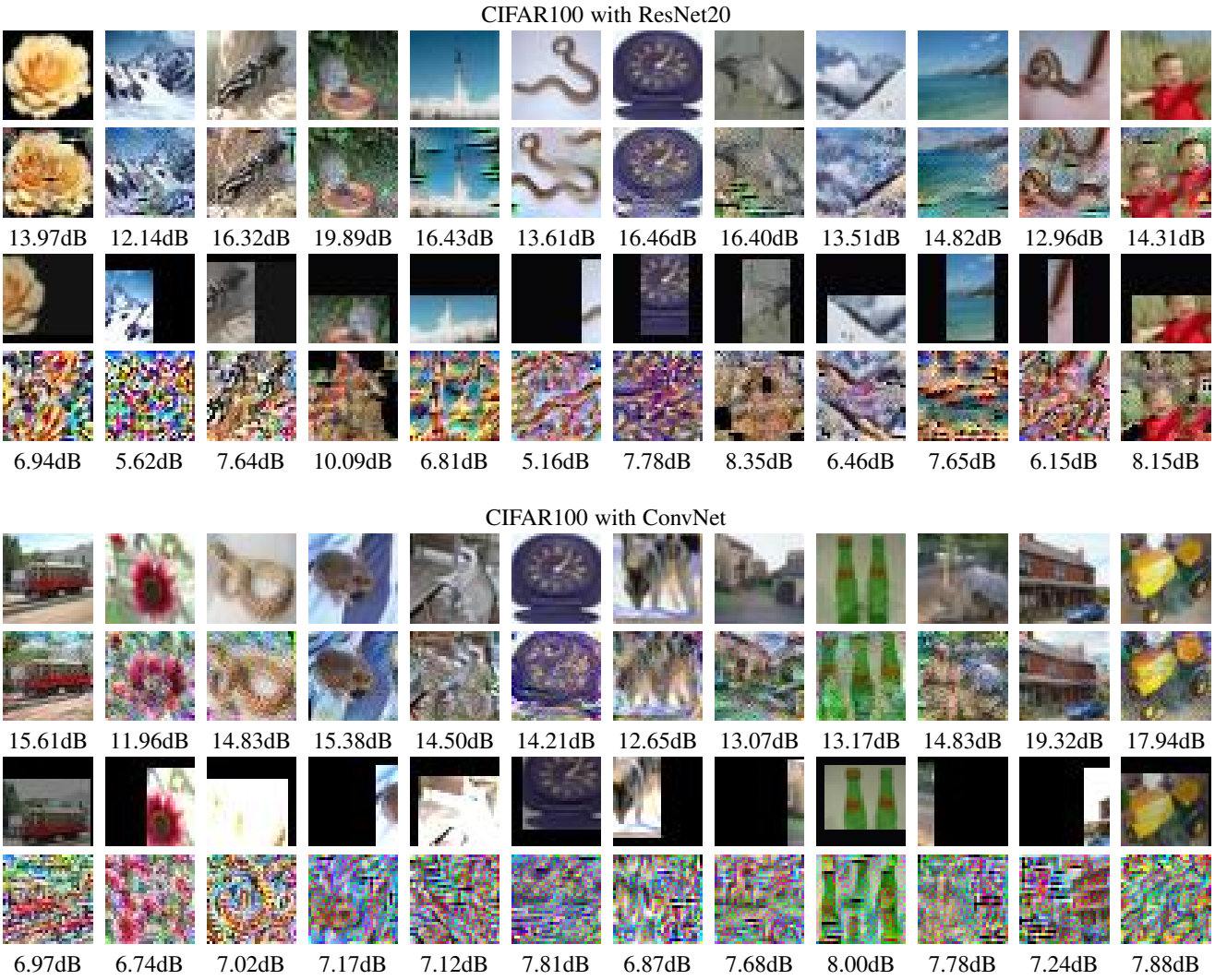


Fig. 8: Visual results and the PSNR values of the reconstruction attacks [13] with and without our defense on CIFAR100. Row 1: clean samples, Row 2: reconstructed samples without transformation, Row 3: transformed samples, Row 4: reconstructed samples with transformation. The adopted transformations are the corresponding *Hybrid* policies in Table 2.

final model. Compared with our earlier work, our hybrid policies achieve superior performance in preventing privacy leakage while preserving similar model performance.

Table 3 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, differential privacy with different noise scales and types, and mixing different number of images. Table 4 illustrates the comparison results. We observe that both differential privacy and pruning techniques can hardly reduce the PSNR values, and the results are consistent with the conclusion in [11]. We also compare the variance of their PSNR values, which indicates the defense performance of adding noise or pruning parameters varies in different classes and images. Although InstaHide can achieve similar defense performance on different classes and samples, the model accuracy is decreased significantly and the PSNR values show that InstaHide cannot resist state-of-the-art reconstruction attacks. In contrast, our solution can significantly destruct the quality of recovered images, while maintaining high model accuracy and low sample variance.

Comparisons with dataset condensation. We further compare the performance of our method with the dataset condensation approach, which is used for efficient and privacy-preserving training. In particular, we implement several state-of-the-art dataset condensation methods including GS-WGAN [43], DP-MERF [54], DC [44], DSA [45], and DM [47]. For GS-WGAN and DP-MERF, we follow their default privacy settings: $\epsilon = 10$ for GS-WGAN and $\epsilon = 1$ for DP-MERF. We set the parameter ipc (images per class) as 50. We train the shared model on the synthetic datasets and evaluate the model performance on the original test dataset. As shown in Table 5, most dataset condensation methods suffer from both low model performance and high privacy leakage. Although DM achieves a low PSNR value, its accuracy performance is unacceptable. In contrast, our method can preserve both sample

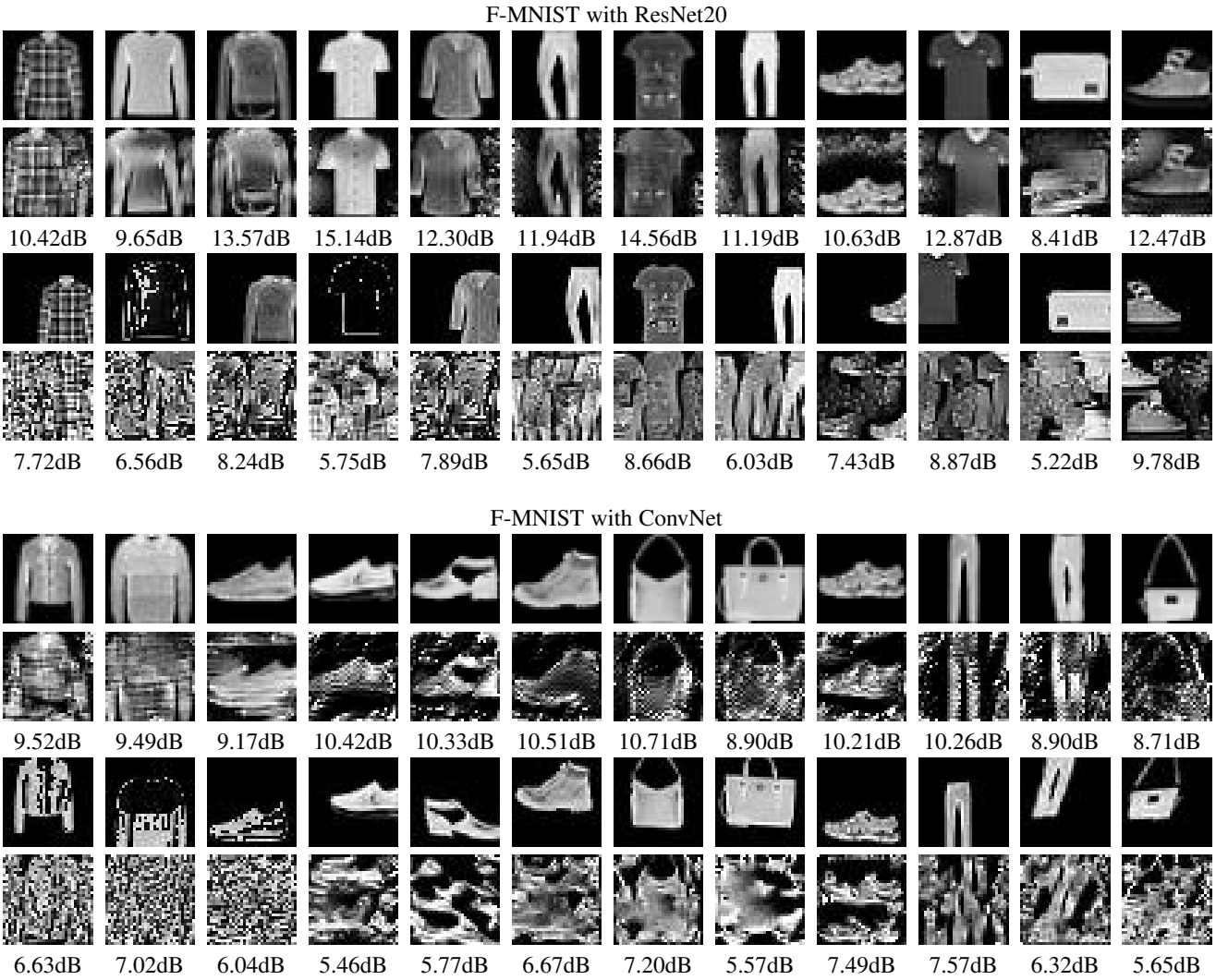


Fig. 9: Visual results and the PSNR values of the reconstruction attacks [13] with and without our defense on F-MNIST.

privacy and model accuracy.

5.4 Ablation Study

Variance. We analyze how the integrated variance term in the privacy score helps the search algorithm to find effective policies. As shown in Table 6, we compare the mean and variance of PSNR of the top-2 and hybrid policies selected by the search algorithm with and without Var on the two datasets. The policy chosen by the new privacy score generally has a lower divergence in defeating the reconstruction attacks. With the variance term, the search algorithm would choose policies that have higher privacy protection capability on different class samples. Therefore, the algorithm with our privacy score achieves better performance against the reconstruction attacks. The visual comparison in Fig. 10 also indicates the effectiveness of the variance component in our privacy score.

Combine top k augmentations. A straightforward solution for policy construction is to select the top k basic augmentations and randomly combine them during the training phase. However, simply stacking the top k augmentations would provide a significant negative impact on the model accuracy. As shown in Table 7, we evaluate the 50 basic augmentations on CIFAR100 with ResNet20 and select the top 3 augmentations: 3rd, 43th, and 15th. Although combining top augmentations indeed decreases the PSNR values, model accuracy also suffers an unacceptable decline ($\sim 9\%$).

5.5 Hyperparameter Analysis

Attack cost. We first analyze the impact of attack cost on the extraction results. The attacker can perform more optimization steps to try to recover more accurate images. In our experiments, we increase the total number of attack optimization iterations from 4800 to 7800, 10800, and 13800. Fig. 11a compares the PSNR values without and with our privacy-preserving protection on two models trained over CIFAR100. Fig. 12 visualizes some recovered images. We can observe that when the number of training epochs increases, the PSNR values of our method slightly decrease and are always less than 8.04 (Conv) and 7.64 (ResNet). The enhancement from higher attack cost (nearly $3\times$ computational overhead) is negligible with our policies (less than 1 PSNR improvement).

Training Epochs. Next, we analyze the effectiveness of our proposed method under different model statuses, i.e., the number of training epochs. We train the shared model over CIFAR100

TABLE 2: PSNR (db) and Model Accuracy (%) of Different Transformation Configurations for Each Architecture and Dataset

Policy	PSNR	ACC
Standard	12.11	76.88
Random	11.41	73.94
[15]	7.91	76.89
13-43-18	8.21	76.24
21-3-16	6.83	71.56
Hybrid	7.35	77.25

(a) CIFAR100 with ResNet20

Policy	PSNR	ACC
Standard	13.65	72.68
Random	12.18	69.91
[15]	7.31	68.55
21-13-3	6.21	64.41
15-48-15	7.78	68.05
Hybrid	7.21	68.62

(b) CIFAR100 with ConvNet

Policy	PSNR	ACC
Standard	10.08	95.53
Random	9.23	91.16
[15]	7.58	91.12
31-38-48	8.06	90.69
26-39-40	7.40	90.13
Hybrid	7.48	91.55

(c) F-MNIST with ResNet20

Policy	PSNR	ACC
Standard	9.14	94.25
Random	8.83	90.18
[15]	7.90	90.78
26-11-4	7.29	89.74
15-27-13	7.32	90.10
Hybrid	7.45	90.84

(d) F-MNIST with ConvNet

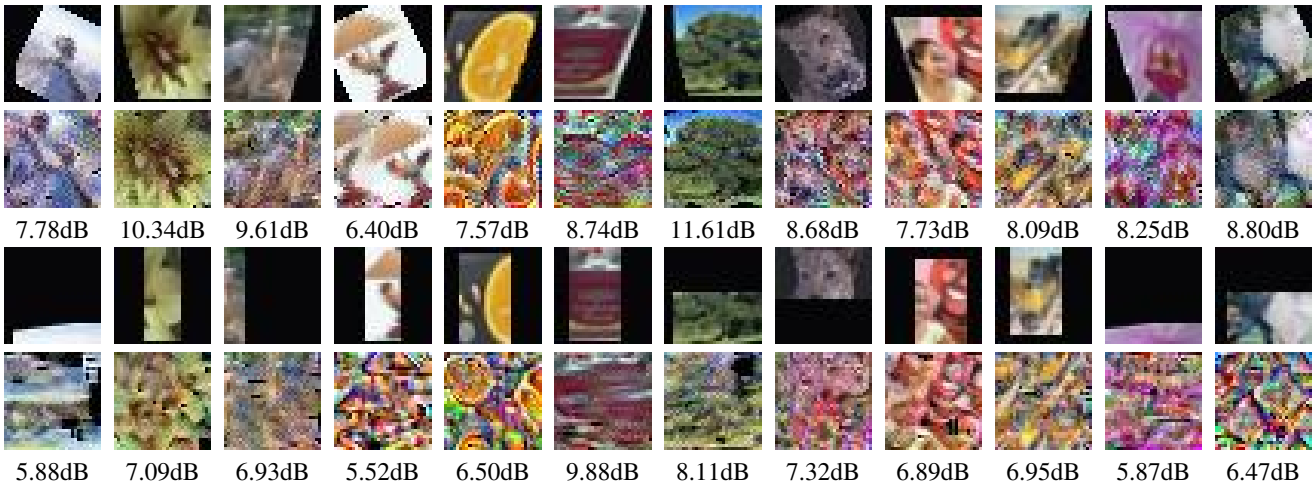
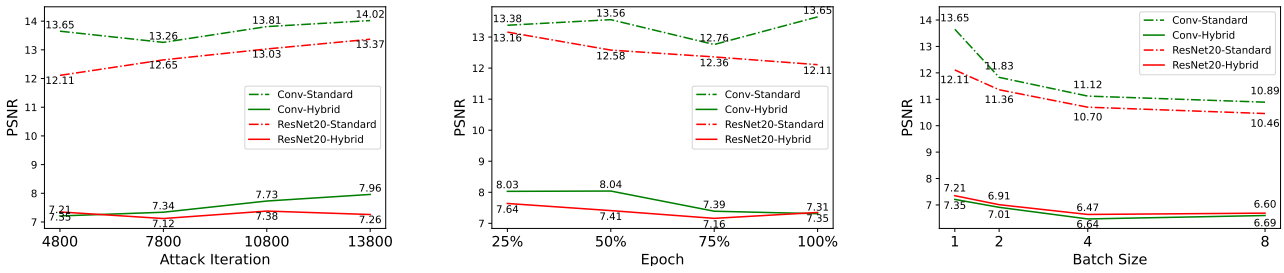


Fig. 10: Visual results and the PSNR values of the reconstruction attacks with the hybrid policies chosen using the privacy scores without (Row 1&2) and with variance (Row 3&4).



(a) Number of iterations (b) Model status (c) Batch size

Fig. 11: Reconstruction results with different numbers of optimization iterations, model status, and batch sizes.

TABLE 3: The PSNR values (db) Between the Reconstructed and Transformed Images under Different Attacks

Attack	Standard	Hybrid	Attack	Standard	Hybrid
LBFGS+L2	8.84	6.31	LBFGS+COS	9.39	6.78
Adam+Cosine	12.11	7.35	Adam+L2	9.23	6.31
Adam+L1	8.78	6.35	SGD+COS	12.36	7.23

with 25%, 50%, 75%, 100% of the total epochs (200 epochs). The defense comparison results are shown in Fig. 11b with the same configurations as the above experiment. We can observe that our privacy-preserving solution always shows high effectiveness to protect the training samples of different model statuses.

Batch size. We consider the impact of the batch size. Generally, it is more difficult for the attacker to reconstruct training samples from the averaged gradients with larger batch sizes. Here, we report the averaged PSNR values when the batch size is $1, 2, 4, 8$ in Fig. 11c. We also provide the visual reconstruction results with each batch size in Fig. 13. We confirm that a larger batch size leads to worse attack results, and our defense is effective even when the batch size is 1.

Architecture. To demonstrate the superiority of our proposed method, we also search privacy policies with MobileNetV2 [68] and ViT [69] on CIFAR100. We train MobileNetV2 from scratch and use the same training hyper-parameters as ResNet20. We fine-tune the ViT model for 10 epochs, which was pretrained on the ImageNet dataset. As shown in Table 8a and 8b, our policies can successfully reduce privacy leakage and maintain model performance. The visual comparison results are also provided in Fig. 14. For MobileNetV2, the reconstructed images are of low quality with the protection of our method. For ViT, the reconstruction

TABLE 4: Comparisons with Existing Defense Methods under the Adam+Cosine Attack

Defense	Acc	PSNR	Var
Standard	76.88	12.11	3.0
[15]	76.89	7.91	3.50
Gaussian(1e-3)	75.75	10.90	2.44
Gaussian(1e-2)	46.64	10.09	1.41
Laplacian(1e-3)	74.68	10.89	2.60
Laplacian(1e-2)	41.7	10.48	1.51
Pruing(0.7)	77.19	10.21	2.64
Pruing(0.95)	70.19	10.01	1.85
Pruing(0.99)	57.43	9.59	1.19
InstaHide(4)	69.12	10.46	0.50
InstaHide(6)	67.53	10.58	0.37
Proposed	77.25	7.35	2.78

TABLE 5: Comparisons with Existing Dataset Condensation Methods

Method	Dataset	PSNR	ACC
Standard	F-MNIST	9.14	94.25
GS-WGAN [43]	F-MNIST-ipc50	12.47	52.68
DP-MERF [54]	F-MNIST-ipc50	11.57	60.13
DC [44]	F-MNIST-ipc50	9.13	39.49
DSA [45]	F-MNIST-ipc50	9.07	50.16
DM [47]	F-MNIST-ipc50	6.24	64.01
Proposed	F-MNIST	7.45	90.84

process is composed of the patch reconstruction and our search policies not only prevent the recovery of a single patch but also disrupt the patch positions.

High-resolution datasets. We further evaluate our method in two high-resolution datasets CelebA (112×112) and ImageNet (224×224) with ResNet18. To better demonstrate the quality of the reconstructed faces, we crop the faces out of the corresponding images and scale them to 112×112. For CelebA, we use 40 attributes classification as our training task, while we take 25 classes from ImageNet to reduce the training cost. As shown in Table 8c and 8d, our search policies can effectively protect the privacy of the training datasets as well as preserve the accuracy of the shared models. The visual results in Fig. 14 also confirm the superiority of our method.

5.6 Transferability Analysis

In the above experiments, we search for 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 with ResNet20 to the tasks of F-MNIST and ImageNet. As shown in Table 9, for F-MNIST, we observe that although these transferred policies are slightly worse than the ones directly searched from the dataset, they are still very effective in preserving privacy and model performance, and better than the randomly selected policies. For ImageNet, our searched policies could also significantly reduce the quality of the recovered images. This transferability property makes our solution more efficient.

TABLE 6: Comparisons Between Policies Chosen from [15] and the Variance Integrated Algorithm on CIFAR100 and F-MNIST with ResNet20

Defense	Policy	PSNR	Var
[15]	3-1-7	6.72	1.61
	43-18-18	9.06	3.13
	Hybrid	7.91	3.50
Proposed	13-43-18	8.21	1.75
	21-3-16	6.83	1.15
	Hybrid	7.35	2.78

(a) CIFAR100

Defense	Policy	PSNR	Var
[15]	19-15-45	7.21	0.59
	2-43-21	7.48	1.77
	Hybrid	7.58	1.45
Proposed	31-38-48	8.06	2.81
	26-39-40	7.40	0.48
	Hybrid	7.48	1.21

(b) F-MNIST

TABLE 7: Comparisons with Top 3 Augmentations

Policy	PSNR	ACC
Standard	12.11	76.88
{3}	6.94	71.11
{3,43}	6.40	69.81
{3,43,15}	5.95	67.91
Proposed	7.35	77.25

5.7 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. Actually, the general adaptive attack towards our privacy-preserving method is not easy to design, we adopt a simple way to explore the potential risk of adaptive attacks. We suppose the adversary can possibly initialize the starting image x_0 as black pixels when the shift transformation is applied. As shown in Table 10, our solution still achieves a good privacy protection capability even when the adversary obtains part of image information.

5.8 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. Fig. 15 shows the privacy score of each transformation. The five transformations with the lowest scores are (red bars in the figure): 3rd [horizontal shifting, 9], 43rd [vertial shifting 5], 15th [brightness, 9], 48[vertical shifting 3], and 14th [sharpness 9], 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 the black area, which can undermine the quality of the recovered image during the optimization. Brightness and sharpness aim to modify the lightness and blurring level of an image. These operations can blur the local details, which also increases the difficulty of image reconstruction. Overall, the selected privacy-preserving transformations can distort the details of the images, while maintaining the semantic information. We illustrate the visual results and PSRN values of the reconstructed images of

TABLE 8: PSNR (db) and model accuracy (%) of different transformation policies with different architectures and datasets

Policy	PSNR	ACC
Standard	11.24	65.96
6-20-42	8.27	63.63
38-15-48	7.51	61.92
Hybrid	7.86	63.03

(a) CIFAR100-MobileNetV2

Policy	PSNR	ACC
Standard	13.21	87.83
20-3-38	8.46	80.06
7-5-21	8.74	85.68
Hybrid	8.63	85.12

(b) CIFAR100-ViT

Policy	PSNR	ACC
Standard	18.03	90.69
21-19	13.92	89.95
3-2-38	14.38	82.06
Hybrid	14.01	89.47

(c) CelebA-ResNet18

Policy	PSNR	ACC
Standard	12.53	82.24
42-21-3	7.62	78.46
24-15-48	8.89	82.16
Hybrid	8.43	84.75

(d) ImageNet-ResNet18

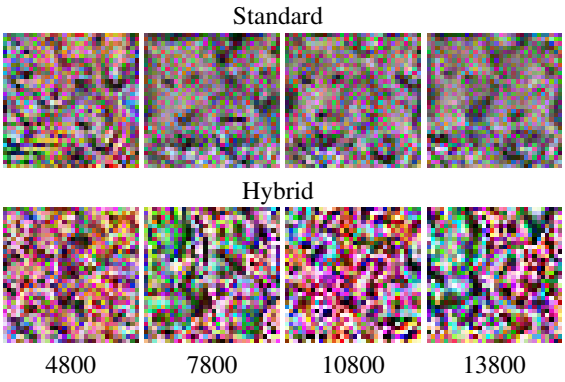


Fig. 12: Visual results of different attack iterations.

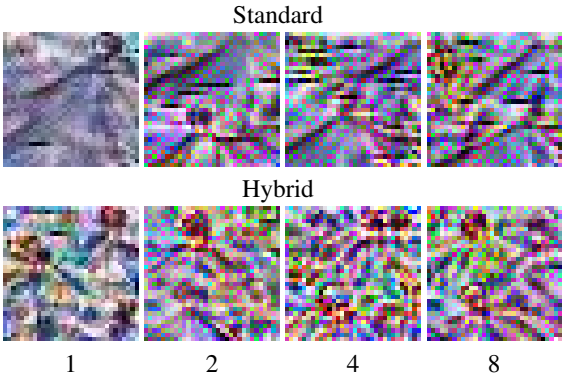


Fig. 13: Visual results of the reconstruction attacks with different batch sizes (1, 2, 4, 8).

four kinds of transformations with better (shift and brightness) and worse (equalize and solarize) privacy protection in Fig. 16. We can observe that with the privacy-preserving transformations (first two columns), massive visual information is lost (fewer edges), which increases the difference between the gradients of the transformed images and the original ones. Besides, the PSNR values of the reconstructed images with shift and brightness are lower than the ones with equalize and solarize. The reason is that the attacks can hardly reconstruct the areas with little visual 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. Fig.17 demonstrates the similarity between 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

TABLE 9: Transferability Results: Applying the Same Policies from CIFAR100 to F-MNIST and ImageNet

Policy	PSNR	ACC
Standard	10.08	95.53
13-43-18	8.37	89.83
21-3-16	7.48	86.64
Hybrid	7.64	90.42

(a) F-MNIST with ResNet20

Policy	PSNR	ACC
Standard	9.12	69.07
13-43-18	5.47	67.24
21-3-16	6.84	66.84
Hybrid	6.95	67.40

(b) ImageNet with ResNet20

TABLE 10: PSNR (db) of Different Initializations for Each Architecture on CIFAR100

Model	Transformation Policy	Initialized to random values	Initialized to black pixels
ResNet20	21-3-16	6.83	6.89
ConvNet	21-13-3	6.21	6.25

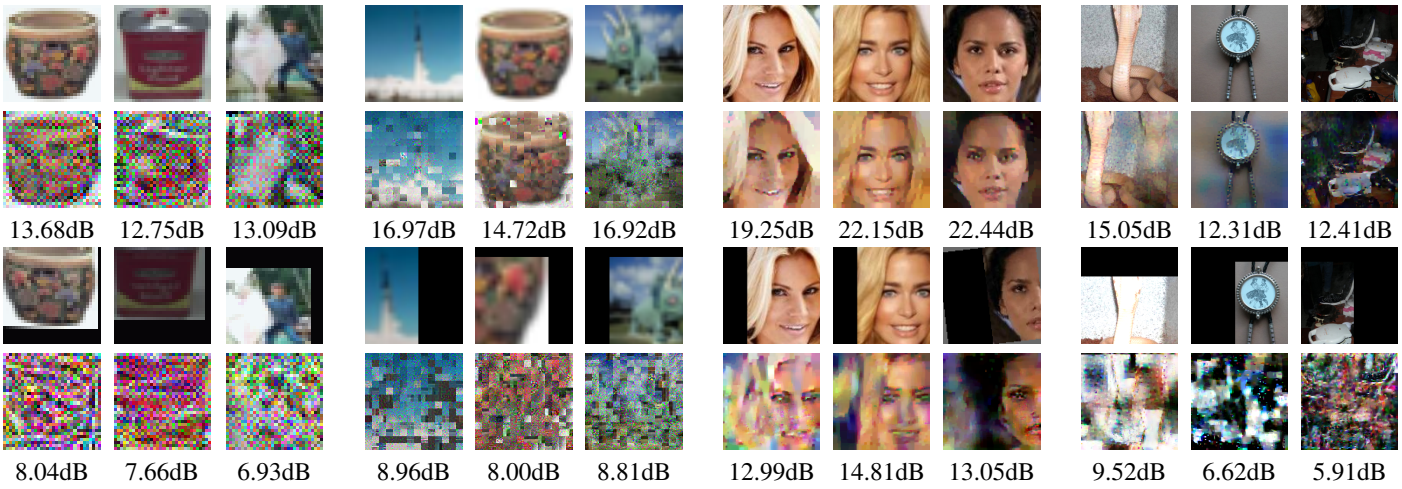
the input, e.g. color, shape, and texture. The similarity scores for all three cases are almost the same at deep layers. This reveals the optimal policy has a negligible impact on the semantic information of the images used for classification, and the model performance is thus maintained.

6 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. Although we focus on the computer vision domain and image classification tasks, the reconstruction attacks may occur in other domains, e.g., natural language processing [11]. Then the searched image transformations cannot be applied. However, it is possible to use text augmentation techniques [74], [75] (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.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work is in part supported by the National Key R&D Program of China under Grant 2022YFB3103500; in part by the National Natural Science Foundation of China under Grants U21A20463, U20A20176, and 62102052; in part by the Singapore National



(a) CIFAR100 with MobileNetV2 (b) CIFAR100 with ViT (c) CelebA with ResNet18 (d) ImageNet with ResNet18

Fig. 14: Visual results and the PSNR values of the reconstruction attacks without (Row 1&2) and with (Row 3&4) our defense under different architectures and high-resolution datasets.

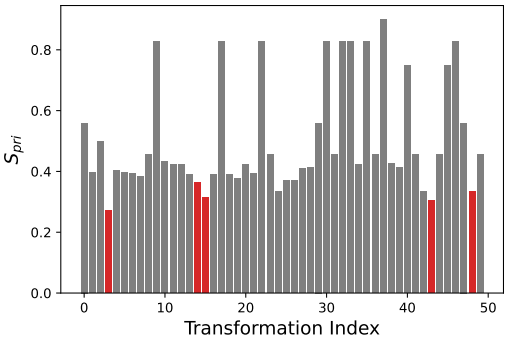


Fig. 15: Privacy scores of the 50 transformation functions in the augmentation library.

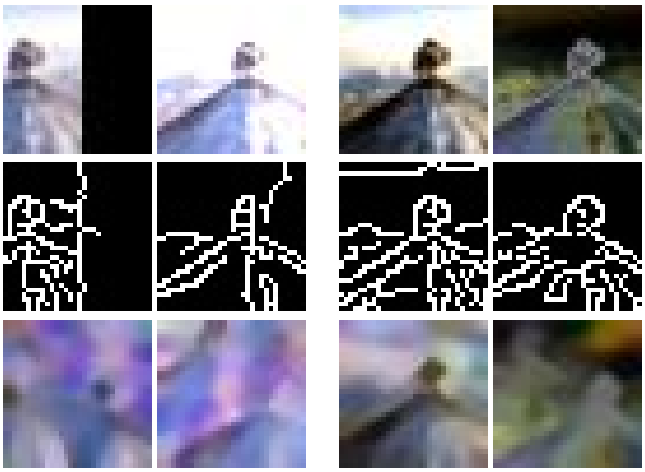


Fig. 16: Visual results and the PSNR values of the reconstruction images with better (first two columns: shift and brightness) and worse (last two columns: equalize and solarize) privacy protection. Row 1-3: transformed images, canny edge detection results of the transformed images [73], and reconstructed images.

Research Foundation under its National Cybersecurity R&D Programme under NCR Award NRF2018NCR-NCR009-0001, in part supported by the Singapore Ministry of Education (MOE) under Grant AcRF Tier 2 MOE-T2EP20121-0006 and Grant AcRF Tier 1 RS02/19; and in part by the Nanyang Technological University (NTU) Start-up grant. This work is also sponsored by CCF-AFSG Research Fund.

REFERENCES

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, 2019.

[2] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *IEEE Symposium on Security and Privacy*, 2019.

[3] S. Guo, T. Zhang, X. Xie, L. Ma, T. Xiang, and Y. Liu, "Towards byzantine-resilient learning in decentralized systems," *arXiv preprint arXiv:2002.08569*, 2020.

[4] S. Guo, X. Zhang, F. Yang, T. Zhang, Y. Gan, T. Xiang, and Y. Liu, "Robust and privacy-preserving collaborative learning: A comprehensive survey," *arXiv preprint arXiv:2112.10183*, 2021.

[5] Z. Wang, H. Xiao, Y. Duan, J. Zhou, and J. Lu, "Learning deep binary descriptors via bitwise interaction mining," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

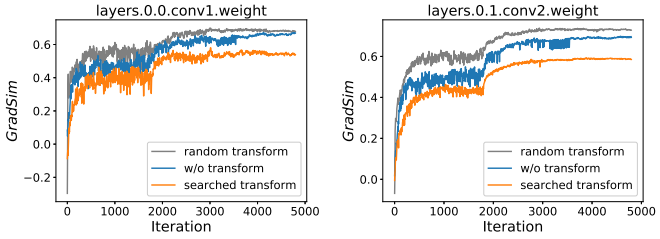
[6] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, 2019.

[7] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, 2020.

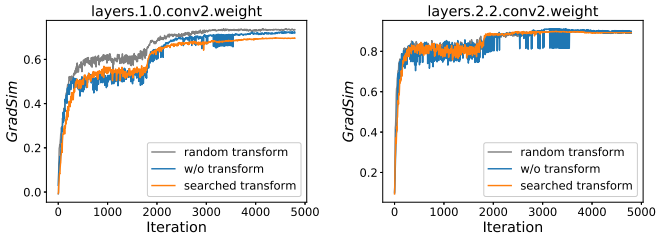
[8] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *International Journal of Medical Informatics*, 2018.

[9] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: information leakage from collaborative deep learning," in *ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[10] M. Nasr, R. Shokri, and A. 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.



(a) Shallow layers



(b) Deep layers

Fig. 17: Gradient similarity during the reconstruction optimization process, for CIFAR100 with ResNet20.

[11] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Adv. Neural Inform. Process. Syst.*, 2019.

[12] B. Zhao, K. R. Mopuri, and H. Bilen, "iDLG: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.

[13] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients–how easy is it to break privacy in federated learning?" in *Adv. Neural Inform. Process. Syst.*, 2020.

[14] W. Wei, L. Liu, M. Loper, K.-H. Chow, M. E. Gursoy, S. Truex, and Y. Wu, "A framework for evaluating gradient leakage attacks in federated learning," *arXiv preprint arXiv:2004.10397*, 2020.

[15] W. Gao, S. Guo, T. Zhang, H. Qiu, Y. Wen, and Y. Liu, "Privacy-preserving collaborative learning with automatic transformation search," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[16] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang *et al.*, "Large scale distributed deep networks," *Adv. Neural Inform. Process. Syst.*, vol. 25, pp. 1223–1231, 2012.

[17] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 6341–6345.

[18] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large cnns at the edge," *arXiv preprint arXiv:2007.14513*, 2020.

[19] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling distributed machine learning with the parameter server," in *USENIX Symposium on Operating Systems Design and Implementation*, 2014, pp. 583–598.

[20] M. Liu, W. Zhang, Y. Mroueh, X. Cui, J. Ross, T. Yang, and P. Das, "A decentralized parallel algorithm for training generative adversarial nets," *arXiv preprint arXiv:1910.12999*, 2019.

[21] T. Sun, D. Li, and B. Wang, "Stability and generalization of the decentralized stochastic gradient descent," *arXiv preprint arXiv:2102.01302*, 2021.

[22] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Int. Conf. Comput. Vis.*, 2019.

[23] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast autoaugment," *Adv. Neural Inform. Process. Syst.*, 2019.

[24] P. Li, X. Liu, and X. Xie, "Learning sample-specific policies for sequential image augmentation," in *ACM Int. Conf. Multimedia*, 2021.

[25] M. Wistuba, A. Rawat, and T. Pedapati, "A survey on neural architecture search," *arXiv preprint arXiv:1905.01392*, 2019.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009.

[27] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

[28] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1-3, 1989.

[29] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[30] J. Jeon, K. Lee, S. Oh, J. Ok *et al.*, "Gradient inversion with generative image prior," *Adv. Neural Inform. Process. Syst.*, 2021.

[31] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," *arXiv preprint arXiv:2203.15696*, 2022.

[32] A. Hatamizadeh, H. Yin, H. R. Roth, W. Li, J. Kautz, D. Xu, and P. Molchanov, "Gradvit: Gradient inversion of vision transformers," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 10 021–10 030.

[33] G. Zhang, B. Liu, H. Tian, T. Zhu, M. Ding, and W. Zhou, "How does a deep learning model architecture impact its privacy?" *arXiv preprint arXiv:2210.11049*, 2022.

[34] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM SIGSAC conference on Computer and Communications Security*, 2016.

[35] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *IEEE Symposium on Security and Privacy*, 2019.

[36] H. Phan, M. T. Thai, H. Hu, R. Jin, T. Sun, and D. Dou, "Scalable differential privacy with certified robustness in adversarial learning," in *International Conference on Machine Learning*, 2020.

[37] S. Guo, T. Zhang, T. Xiang, and Y. Liu, "Differentially private decentralized learning," *arXiv preprint arXiv:2006.07817*, 2020.

[38] Q. Zhao, C. Zhao, S. Cui, S. Jing, and Z. Chen, "PrivateDL: Privacy-preserving collaborative deep learning against leakage from gradient sharing," *International Journal of Intelligent Systems*, 2020.

[39] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for federated learning on user-held data," *arXiv preprint arXiv:1611.04482*, 2016.

[40] L. Fan, K. W. Ng, C. Ju, T. Zhang, C. Liu, C. S. Chan, and Q. Yang, "Rethinking privacy preserving deep learning: How to evaluate and thwart privacy attacks," *arXiv preprint arXiv:2006.11601*, 2020.

[41] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. Chen, "Soteria: Provable defense against privacy leakage in federated learning from representation perspective," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[42] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "Gan-leaks: A taxonomy of membership inference attacks against generative models," in *ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 343–362.

[43] D. Chen, T. Orekondy, and M. Fritz, "Gs-wgan: A gradient-sanitized approach for learning differentially private generators," *Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 12 673–12 684, 2020.

[44] B. Zhao, K. R. Mopuri, and H. Bilen, "Dataset condensation with gradient matching." *Int. Conf. Learn. Represent.*, 2021.

[45] B. Zhao and H. Bilen, "Dataset condensation with differentiable siamese augmentation," in *International Conference on Machine Learning*, 2021, pp. 12 674–12 685.

[46] T. Dong, B. Zhao, and L. Lyu, "Privacy for free: How does dataset condensation help privacy?" *arXiv preprint arXiv:2206.00240*, 2022.

[47] B. Zhao and H. Bilen, "Dataset condensation with distribution matching," 2023.

[48] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Pattern Recognition*, 2010.

[49] H. Qiu, Y. Zeng, T. Zhang, Y. Jiang, and M. Qiu, "Fencebox: A platform for defeating adversarial examples with data augmentation techniques," *arXiv preprint arXiv:2012.01701*, 2020.

[50] H. Qiu, Y. Zeng, Q. Zheng, T. Zhang, M. Qiu, and G. Memmi, "Mitigating advanced adversarial attacks with more advanced gradient obfuscation techniques," *arXiv preprint arXiv:2005.13712*, 2020.

[51] H. Qiu, Q. Zheng, T. Zhang, M. Qiu, G. Memmi, and J. Lu, "Towards secure and efficient deep learning inference in dependable iot systems," *IEEE Internet of Things Journal*, 2020.

[52] Y. Zeng, H. Qiu, S. Guo, T. Zhang, M. Qiu, and B. Thuraisingham, "Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation," in *arXiv preprint arXiv:2012.07006*, 2021.

[53] S. Guo, T. Zhang, H. Qiu, Y. Zeng, T. Xiang, and Y. Liu, "The hidden vulnerability of watermarking for deep neural networks," *arXiv preprint arXiv:2009.08697*, 2020.

[54] F. Harder, K. Adamczewski, and M. Park, "Dp-merf: Differentially private mean embeddings with randomfeatures for practical privacy-preserving data generation," in *International Conference on Artificial Jntelligence and Statistics*, 2021, pp. 1819–1827.

[55] N. Lee, T. Ajanthan, and P. H. Torr, "Snip: Single-shot network pruning based on connection sensitivity," *arXiv preprint arXiv:1810.02340*, 2018.

[56] C. Wang, G. Zhang, and R. Grosse, "Picking winning tickets before training by preserving gradient flow," *arXiv preprint arXiv:2002.07376*, 2020.

[57] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," *Adv. Neural Inform. Process. Syst.*, 2020.

[58] J. Mellor, J. Turner, A. Storkey, and E. J. Crowley, "Neural architecture search without training," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7588–7598.

[59] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, 2011.

[60] P. Popien, "AutoAugment - Learning Augmentation Policies from Data," https://github.com/DeepVoltaire/AutoAugment.

[61] R. M. Neal, "Annealed importance sampling," *Statistics and Computing*, 2001.

[62] N. M. Kwok, G. Fang, and W. Zhou, "Evolutionary particle filter: Re-sampling from the genetic algorithm perspective," in *International Conference on Intelligent Robots and Systems*, 2005.

[63] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[64] M. Balunović, D. I. Dimitrov, R. Staab, and M. Vechev, "Bayesian framework for gradient leakage," *arXiv preprint arXiv:2111.04706*, 2021.

[65] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[66] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[68] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4510–4520.

[69] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda, "Visual transformers: Token-based image representation and processing for computer vision," *arXiv preprint arXiv:2006.03677*, 2020.

[70] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3730–3738.

[71] D. Aritra, H. B. El, M. A. Ahmed, H. Chen-Yu, N. S. Atal, C. Marco, and K. Panos, "On the discrepancy between the theoretical analysis and practical implementations of compressed communication for distributed deep learning," in *AAAI*, 2019.

[72] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inform. Process. Syst.*, 2019.

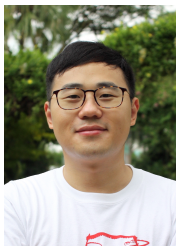
[73] Z. Xu, X. Baojie, and W. Guoxin, "Canny edge detection based on open cv," in *2017 13th IEEE international conference on electronic measurement & instruments (ICEMI)*. IEEE, 2017, pp. 53–56.

[74] S. Kobayashi, "Contextual augmentation: Data augmentation by words with paradigmatic relations," *arXiv preprint arXiv:1805.06201*, 2018.

[75] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

Gao Wei received the BS degree from Beihang University, Beijing China in 2019. He is currently pursuing the PhD degree at Nanyang Technological University, Singapore. His research interests include distributed machine learning system and machine learning security.

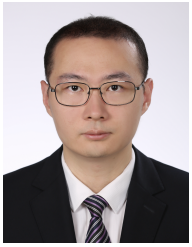
Xu Zhang received the B.E. degree from Chongqing University, China in 2021. He is currently working toward the Master degree at Chongqing University. His research interests include machine learning security and distributed computing security.



Shangwei Guo is an associate professor in College of Computer Science, Chongqing University. He received the Ph.D. degree in computer science from Chongqing University, Chongqing, China at 2017. He worked as a postdoctoral research fellow at Hong Kong Baptist University and Nanyang Technological University from 2018 to 2020. His research interests include machine learning security, cloud/edge computing security, and database security.

Tianwei Zhang is an assistant professor in School of Computer Science and Engineering, at Nanyang Technological University. His research focuses on computer system security. He is particularly interested in security threats and defenses in machine learning systems, autonomous systems, computer architecture and distributed systems. He received his Bachelor's degree at Peking University in 2011, and the Ph.D degree in at Princeton University in 2017.

Tao Xiang received the BEng, MS and PhD degrees in computer science from Chongqing University, China, in 2003, 2005, and 2008, respectively. He is currently a Professor of the College of Computer Science at Chongqing University. Prof. Xiang's research interests include multimedia security, cloud security, data privacy and cryptography. He has published over 100 papers on international journals and conferences. He also served as a referee for numerous international journals and conferences.



Han Qiu received the B.E. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2011, the M.S. degree from Telecom-ParisTech (Institute Eurecom), Biot, France, in 2013, and the Ph.D. degree in computer science from the Department of Networks and Computer Science, Telecom-Paris, Paris, France, in 2017. He worked as a postdoc and a research engineer with Telecom Paris and LINCS Lab from 2017 to 2020. Currently, he is an assistant professor at Institute for Network Sciences and Cyberspace, Tsinghua University, China. His research interests include AI security, data security, and cloud computing.

Yonggang Wen is the Professor of Computer Science and Engineering at Nanyang Technological University (NTU), Singapore. He has also served as the Associate Dean (Research) at College of Engineering at NTU Singapore since 2018. He received his PhD degree in Electrical Engineering and Computer Science from Massachusetts Institute of Technology (MIT), Cambridge, USA, in 2008. He was with Cisco, San Jose, CA, USA, where he led product development in content delivery network, which had a revenue impact of 3 Billion US dollars globally. His work in Multi-Screen Cloud Social TV has been featured by more than 1600 news articles from over 29 countries and received 2013 ASEAN ICT Awards (Gold Medal). His work on Cloud3DView, as the only academia entry, has won 2016 ASEAN ICT Awards (Gold Medal) and 2015 Datacentre Dynamics Awards 2015. He serves on editorial boards for multiple transactions and journals, including IEEE Transactions on Circuits and Systems for Video Technology, IEEE Wireless Communication Magazine, IEEE Communications Survey & Tutorials, IEEE Transactions on Multimedia, etc. His research interests include cloud computing, green data center, distributed machine learning, blockchain, big data analytics, multimedia network and mobile computing. He is a Fellow of IEEE, and a Distinguished Member of ACM.

Yang Liu received the B.Comp. degree (Hons.) from the National University of Singapore (NUS) in 2005 and the Ph.D. degree from NUS and MIT, in 2010. He started his postdoctoral work in NUS and MIT. In 2012, he joined Nanyang Technological University (NTU). He is currently a Full Professor and the Director of the Cybersecurity Laboratory, NTU. He specializes in software verification, security, and software engineering. His research has bridged the gap between the theory and practical usage of formal methods and program analysis to evaluate the design and implementation of software for high assurance and security. By now, he has more than 270 publications in top tier conferences and journals. He received a number of prestigious awards, including the MSRA Fellowship, the TRF Fellowship, the Nanyang Assistant Professor, the Tan Chin Tuan Fellowship, the Nanyang Research Award, and eight best paper awards in top conferences, such as ASE, FSE, and ICSE.