Gradient Obfuscation Gives a False Sense of Security in Federated Learning

Kai Yue,[1] Richeng Jin,[2] Chau-Wai Wong,[1] Dror Baron,[1] and Huaiyu Dai[1]
[1]North Carolina State University; {kyue, chauwai.wong, dzbaron, hdai}@ncsu.edu
[2]Zhejiang University; richengjin@zju.edu.cn

Abstract

Federated learning has been proposed as a privacy-preserving machine learning framework that enables multiple clients to collaborate without sharing raw data. However, client privacy protection is not guaranteed by design in this framework. Prior work has shown that the gradient sharing strategies in federated learning can be vulnerable to data reconstruction attacks. In practice, though, clients may not transmit raw gradients considering the high communication cost or due to privacy enhancement requirements. Empirical studies have demonstrated that gradient obfuscation, including intentional obfuscation via gradient noise injection and unintentional obfuscation via gradient compression, can provide more privacy protection against reconstruction attacks. In this work, we present a new reconstruction attack framework targeting the image classification task in federated learning. We show how commonly adopted gradient postprocessing procedures, such as gradient quantization, gradient sparsification, and gradient perturbation may give a false sense of security in federated learning. Contrary to prior studies, we argue that privacy enhancement should not be treated as a byproduct of gradient compression. Additionally, we design a new method under the proposed framework to reconstruct images at the semantic level. We quantify the semantic privacy leakage and compare it with conventional image similarity scores. Our comparisons challenge the image data leakage evaluation schemes in the literature. The results emphasize the importance of revisiting and redesigning the privacy protection mechanisms for client data in existing federated learning algorithms.

1 Introduction

The past decade has seen a growing demand for a large amount of training data for deep learning, as well as the increased storage and computational capabilities of edge devices. Federated learning has emerged as a privacy-preserving framework under which multiple participants jointly solve a machine learning problem [34, 58]. In a classical federated network, a central server broadcasts a global model to selected clients and collects model updates without directly accessing raw data. Generic solutions such as federated averaging (FedAvg) [43] have been proposed with different variants. In FedAvg, a server transmits a global model to participants based on a predefined client sampling strategy. On the client side, the model will be locally optimized with decentralized private training data. Once the models are transmitted back to the server, an updated global model is constructed by averaging all received local models. During the whole process, raw data will not be exchanged. To reduce the communication cost and improve the model accuracy, especially when clients' data are not independent and identically distributed (IID), gradient quantization and personalized client cost functions have been incorporated into FedAvg [40, 51].

Even without direct access to the raw data, client privacy is not guaranteed in the aforementioned FedAvg framework. Recent studies have shown that client data can be reconstructed based on gradient inversion attack [72, 73]. Specifically, Zhu et al. [73] proposed an attack method that can reconstruct pixel-wise accurate images and token-wise matched texts. This method was further improved to recover high-resolution images with increased batch sizes [17, 65]. Meanwhile, various attack evaluation frameworks have been proposed to quantify the privacy loss based on image similarity scores calculated between the reconstructed images and clients' raw images [26, 61].

To achieve provable differential privacy, researchers have studied adding noise to gradients without significantly reducing the model utility [60, 66]. Nevertheless, it is not clear how differentially private gradients can defend against well-designed data reconstruction attacks [71]. In addition to gradient perturbation, empirical studies demonstrated that gradient compression can serve as a good deterrence to the reconstruction attack [61, 73]. Recent studies suggest that low-bit gradient quantization can be used as effective defenses [70]. Since gradient compression is proposed to reduce the communication cost, privacy protection is not considered by design. Quantifying the privacy gain of various gradient obfuscation

procedures, including intentional ones such as differential privacy through noise injection and unintentional ones such as gradient compression, remains an open problem.

In this study, we show that the aforementioned gradient obfuscation strategies may give a false sense of security. We investigate the image classification task with high-resolution input and improve the image reconstruction attack. We summarize our contributions as follows:

- We propose a reconstruction attack framework with improved reconstruction quality in federated learning. Contrary to previous studies, we show that gradient compression, such as quantized stochastic gradient descent (QSGD) [4] and Top-k sparsification [3], may not be treated as effective defenses to prevent privacy leakage.

- We evaluate several existing defense schemes that perturb the model updates or features. The attack results show that these defenses can still be vulnerable to adversaries, thus advocating the importance of more effective privacy protection designs in federated learning.

- We explore an image reconstruction attack at a semantic level and measure the corresponding image privacy leakage. Although the reconstructions do not necessarily match the original images pixel by pixel, the private information can still be exploited by adversaries. Our work is a step toward a more thorough understanding of the privacy leakage problem in federated learning.

The rest of this paper is organized as follows. Section 2 reviews the relevant work. Section 3 describes the background of federated learning, including the concepts, gradient post-processing tools, and data reconstruction attacks. We present the reconstruction framework in Section 4 and evaluate three existing defense schemes in Section 5. Section 6 introduces the attack at the semantic level and the corresponding evaluation metric. Finally, we conclude the paper and discuss the future work in Section 7.

2 Related Work

2.1 Federated Learning and Client Privacy

In federated learning, a server coordinates the learning procedure and generally has access to the model structure and parameters. In this work, the investigated threat model is an *honest-but-curious* server that follows the FedAvg framework to aggregate client updates, while inspecting the private information without interfering with the training. The attacks concerning data privacy and model confidentiality in federated learning can be categorized into three types: membership inference attack, model inversion/data reconstruction attack, and property inference attack [32, 52].

In spite of its popularity, FedAvg and its variants have exhibited vulnerabilities in protecting client data privacy [42].

Recent studies have shown a particular interest in model inversion attacks. Zhu et al. [73] proposed the *deep leakage from gradient* (DLG) algorithm, in which a dummy input batch of data is updated iteratively to match the shared gradients. They demonstrated that DLG could reconstruct pixel-wise accurate images and token-wise matched texts. This attack has been further developed by revealing labels via analytical methods [57, 65, 72]. Researchers have proposed to reconstruct high-resolution inputs by employing the image prior and improving the design of the cost functions [17, 31, 65]. For example, Jeon et al. [31] updated the parameters of a generative model to obtain reconstructed results. Meanwhile, it has been shown that the modern Transformer model adopted in language tasks can also be compromised under the data reconstruction attack [15]. From a mathematical viewpoint, researchers have demonstrated that DLG is equivalent to solving a system of equations [49, 50]. The data reconstruction attack can also be interpreted from a Bayesian perspective [6].

Our work is closely related to those studies focused on gradient inversion attacks in federated learning [17, 65, 72, 73]. We propose a new attack framework toward high-resolution image data reconstruction in federated learning. By assuming a successful analytical label recovery and improving the design to reduce the redundancy in the unknowns, we increase the quality of reconstructed images and the efficiency of the attack algorithm. Compared to prior work that does not consider client local update [65] or does not consider a batch of input data for high-resolution images [17, 73], we show that an adversary can still reconstruct client data in a more realistic federated learning setting.

2.2 Attack Evaluation and Privacy Leakage

In the literature, various evaluation metrics are adopted as proxies to measure privacy leakage. Pioneering work [17, 61, 73] used a conventional image similarity score or the distance between original and reconstructed images as the evaluation metric, including the mean squared error (MSE), peak signal-to-noise ratio (PSNR) [20], and structural similarity index measure (SSIM) [59]. In addition, Wei et al. [61] proposed the attack success rate, namely, the percentage of successfully reconstructed training data. Follow-up studies [26, 65] further leveraged the neural network based image perceptual scores, such as learned perceptual image patch similarity (LPIPS) [69] as the attack evaluation metric. LPIPS is known to emulate humans' perception well so it has been increasingly popular for evaluating the quality of reconstructed images. Compared to other indicators, such as differential privacy budget of the algorithm or mutual information between the training examples and gradient [41, 45], LPIPS is more intuitive and interpretable. We will use LPIPS as one of the evaluation metrics.

In this work, we demonstrate how an attacker can reveal the high-level semantics in the raw image, which may not

be captured by the image similarity scores such as SSIM or PSNR. We find that there exists ambiguity in telling whether an attack is successful or not. These results raise concerns about the existing evaluations and highlight the importance to revisit the privacy leakage issue in federated learning. We further propose a method to evaluate semantic privacy loss.

2.3 Privacy Enhancement and Defense

Preserving privacy is a priority in designing federated learning algorithms. Homomorphic encryption provides privacy protection by aggregating the ciphertexts directly [18]. However, the additional computational cost and communication overhead could dominate the training procedure [67]. Meanwhile, differentially private machine learning has been developed to achieve provable privacy guarantees [1]. After clipping the gradient and adding Gaussian noise, user-level membership privacy can be enhanced at the cost of the model accuracy [60, 62]. More sophisticated schemes add noise to the manifold representation of gradients [35, 66], where the model accuracy can be greatly improved given the same privacy protection with reduced noise power. However, the vulnerabilities can still be exploited by model inversion attacks as differential privacy does not imply protection against reconstruction attacks or Bayesian restoration targeting attribute privacy [22, 71]. Similar issues also exist in those defense methods that put attention on a particular type of attack [10]. Other works focused on the protection provided by a trusted shuffling server [13, 19] or secure aggregation protocols [9]. These methods preserve privacy while increasing the system complexity drastically.

In this study, we perform attacks on various gradient obfuscation defense schemes, including gradient compression [61], differentially private training [62], and representation perturbation [66]. We do not consider cryptography based defense [67] and other more advanced solutions, including blockchain-based decentralized optimization or federated network topology modification [47].

3 Preliminaries

Consider a federated learning architecture optimized with FedAvg, which is a backbone of commonly-adopted federated learning algorithms. Denote the mth client's dataset by $\mathcal{D}_m = \{(\mathbf{x}_{m,i}, y_{m,i})\}_{i=1}^{N_m}$, where the ith example $(\mathbf{x}_{m,i}, y_{m,i})$ contains an input-output pair drawn from a distribution $\mathcal{P}_m$. The local objective function f_m is defined as the empirical risk over the local data:

$$f_m(\mathbf{w}) \triangleq \frac{1}{N_m} \sum_{i=1}^{N_m} \ell(\mathbf{w}; \mathbf{x}_{m,i}, y_{m,i}), \qquad (1)$$

where ℓ is a sample-wise loss function quantifying the error of the model with a weight vector $\mathbf{w} \in \mathbb{R}^d$ estimating the

label $y_{m,i}$ given an input $\mathbf{x}_{m,i}$. Federated learning optimizes the following problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w}) = \frac{1}{M} \sum_{m=1}^{M} f_m(\mathbf{w}), \qquad (2)$$

where M is the total number of clients. In FedAvg, the server will select a subset $\mathcal{C}_k$ of clients and broadcast a global model $\mathbf{w}^{(k)}$ in each communication round k. Once the model is received, the mth client will initialize a local model $\mathbf{w}_m^{(k,0)}$ and optimize it with multiple gradient descent steps. For the gradient descent based method, the local model updated at step t may be formulated as

$$\mathbf{w}_m^{(k,t)} = \mathbf{w}_m^{(k,0)} - \frac{\eta}{N_m} \sum_{u=0}^{t-1} \sum_{i=1}^{N_m} \nabla \ell(\mathbf{w}_m^{(k,u)}; \mathbf{x}_{m,i}, y_{m,i}), \qquad (3)$$

where η is the learning rate. For simplicity, in this work we also call the weight difference $\mathbf{w}_m^{(k,0)} - \mathbf{w}_m^{(k,t)}$ the gradient. After τ local update steps, the client will choose to transmit an obfuscated gradient $\widetilde{\Delta}^{(k)}$ based on a postprocessing function $\phi(\cdot)$, i.e.,

$$\widetilde{\Delta}_m^{(k)} = \phi(\mathbf{w}_m^{(k,0)} - \mathbf{w}_m^{(k,\tau)}). \qquad (4)$$

Threat model. We consider an honest-but-curious server as an attacker. The goal of the attacker is to inspect sensitive information of client private data without accessing them directly. Specifically, the server may reconstruct training examples that are close to raw ones. The attacker knows the shared model $\mathbf{w}^{(k)}$ and transmitted gradients $\widetilde{\Delta}_m^{(k)}$'s, while it does not modify the model or gradients. It can also access public data and pretrained neural network models such as an image-denoising network. In general, the attacker has sufficient computational resources and memory. We do not consider a stronger attacker that can modify the gradients or model weights. It has been shown that when the integrity of the model/gradient is compromised, the data can be trivially reconstructed [16, 63].

We characterize the postprocessing function $\phi(\cdot)$ as follows. First, we describe several examples of $\phi(\cdot)$ that are designed to improve communication efficiency.

Example 1 *(SignSGD Compression). To save communication cost, the client will use a compression function to reduce the size of gradients. For example, one can use the low precision quantization such as signSGD [8], i.e.,*

$$\phi_{sign}(\mathbf{g}) = \text{sign}(\mathbf{g}). \qquad (5)$$

Other commonly adopted compression schemes include Fed-PAQ [51] and Top-k sparsification [3]. We give the example of gradient quantization and sparsification as follows. We use ϕ_q and ϕ_{qsgd} to represent a deterministic uniform quantizer and a stochastic QSGD [4] quantizer, respectively.

Example 2 *(Gradient Quantization). A client can use a quantizer to compress the gradients. Given an input $\mathbf{g}$, a quantization operator Q will process each entry g_i as follows:*

$$Q(g_i) = \frac{\kappa}{s} \|\mathbf{g}\|_p \cdot \text{sign}(g_i) \cdot \xi_i(\mathbf{g}, s), \quad (6)$$

where κ is a scaling factor, s is a predefined parameter and the number of representation levels is equal to $2s+1$, $\|\mathbf{g}\|_p$ is the ℓ_p norm of the input vector $\mathbf{g}$, and $\xi_i(\mathbf{g}, s)$ is an integer value representing the unsigned quantized level of the ith coordinate of the input vector $\mathbf{g}$. For a stochastic quantizer ϕ_{qsgd}, we set ξ_i in (6) by ξ_i^s defined as follows:

$$\xi_i^s(\mathbf{g}, s) = \begin{cases} l & \text{with prob. } 1 - p_i, \\ l+1 & \text{with prob. } p_i = \frac{s|e_i|}{\kappa \|\mathbf{g}\|_p} - l, \end{cases} \quad (7)$$

where $l \in [0, s)$ is an integer and $\frac{|e_i|}{\kappa \|\mathbf{g}\|_p} \in [l/s, (l+1)/s]$.

Example 3 *(Gradient Sparsification). In Top-k compression [3], the client selects the k largest components of the gradient in absolute values and zeros out other entires. We denote this function by ϕ_{topk}.*

Next, we state the definition of differential privacy and give the example of noisy gradients.

Definition 1 *(Differential Privacy). A randomized mechanism $\mathcal{M}: \mathcal{D} \rightarrow \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if, for any two adjacent inputs $d, d' \in \mathcal{D}$ and for any subset of outputs $S \subseteq \mathcal{R}$ it holds that*

$$\mathbb{P}[\mathcal{M}(d) \in S] \leq e^\varepsilon \mathbb{P}[\mathcal{M}(d') \in S] + \delta. \quad (8)$$

Example 4 *(Noisy Gradient). To achieve a provable differential privacy guarantee, the client will inject additive Gaussian noise to the gradient [1, 60]. Suppose the input $\mathbf{g}$ is bounded, the postprocessing function may be written as*

$$\phi_{dp}(\mathbf{g}) = \mathbf{g} + \mathbf{n}, \quad (9)$$

where each entry of $\mathbf{n}$ is an independent Gaussian variable. The privacy protection increases with the noise power [62].

We now review the data reconstruction attack methods of DLG [73] and *inverting gradient* (InvertGrad) [17]. Consider a client m holding image data $\{\mathbf{X}_{m,i}\}_{i=1}^{N_m}$, where each matrix $\mathbf{X}_{m,i} \in \mathbb{R}^{h \times w}$ denotes an image with height h and width w. Here, we assume grayscale images for the simplicity of presentation unless otherwise specified. An adversarial server is interested in reconstructing $\{\hat{\mathbf{X}}_{m,i}\}_{i=1}^{N_m}$ that is close to the client raw data [1]. In DLG [73], the attacker starts from some

---

input $\mathbf{X}'_{m,i}$, calculates the dummy gradient $\Delta'^{(k)}_m$, and solves the following optimization problem to match with the shared gradient $\widetilde{\Delta}^{(k)}_m$:

$$\{\hat{\mathbf{X}}_{m,i}\}_{i=1}^{N_m} = \underset{\{\mathbf{X}'_{m,i}\}_{i=1}^{N_m}}{\text{argmin}} \|\Delta'^{(k)}_m - \widetilde{\Delta}^{(k)}_m\|^2. \quad (10)$$

In InvertGrad [17], the cost function is replaced by cosine distance and the attack optimization is formulated as follows:

$$\{\hat{\mathbf{X}}_{m,i}\}_{i=1}^{N_m} = \underset{\{\mathbf{X}'_{m,i}\}_{i=1}^{N_m}}{\text{argmin}} - \frac{\langle \Delta'^{(k)}_m, \widetilde{\Delta}^{(k)}_m \rangle}{\|\Delta'^{(k)}_m\| \|\widetilde{\Delta}^{(k)}_m\|} + \frac{1}{N_m} \sum_{i=1}^{N_m} \mathcal{L}_{\text{TV}}(\mathbf{X}'_{m,i}), \quad (11)$$

where $\mathcal{L}_{\text{TV}}$ is the total variation given by

$$\mathcal{L}_{\text{TV}}(\mathbf{X}) = \sum_{i,j} \sqrt{|X_{i+1,j} - X_{i,j}|^2 + |X_{i,j+1} - X_{i,j}|^2}, \quad (12)$$

where $X_{i,j}$ denotes the (i, j)th entry of the matrix $\mathbf{X}$.

In a practical federated learning setting, when the gradients are averaged over multiple training examples and local iterations, the attack optimization problem is still difficult to solve [17, 65]. The effectiveness of these attack methods becomes even more questionable when the gradients are post-processed by the function $\phi(\cdot)$, leaving vague conclusions on the privacy protection provided by federated learning.

4 Reconstruction From Obfuscated Gradient

In this section, we present our framework to **r**econstruct image data from the **o**bfuscated **g**radient (ROG). The schematic is illustrated in Figure 1, and the details are given in the following sections. For the simplicity of presentation, we first discuss the reconstruction attack on a single image $\mathbf{X}_{m,i}$. The method can be applied to the reconstruction of a batch of data.

4.1 Attack Scheme

The proposed attack scheme uses four steps to reconstruct client image data. We will first introduce each step in its general form, and discuss the realizations in Section 4.2.

First, the attacker derives the label $y_{m,i}$ and selects an initial image $\mathbf{X}'_{m,i}$. The image can be randomly initialized, for example, by using independent random variables drawn from a Gaussian distribution or a uniform distribution. The initialization can also be implemented as repetitive patterns or a natural image [61]. After the initialization, instead of looking for the numerical solutions of (10) directly, we introduce an encoding step to preprocess the image first. From the perspective of solving the system of equations, the solution will be easier to determine if the number of unknowns decreases. To reduce the correlations among different coordinates of input
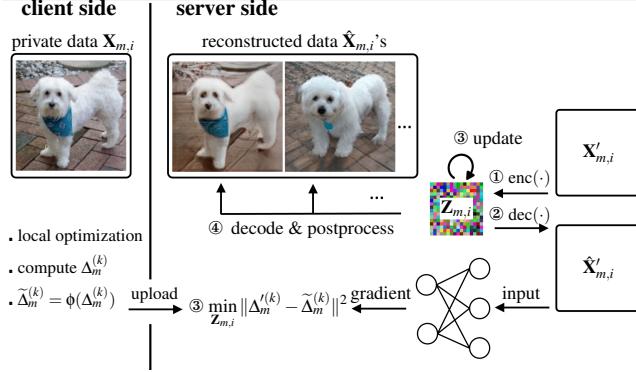
Figure 1: Schematic of the proposed attack method. The attacker first encodes an initial image $\mathbf{X}'_{m,i}$ into a low-dimensional representation $\mathbf{Z}_{m,i}$ to reduce the number of unknowns. Second, the attacker calculates the gradient by decoding the representation $\mathbf{Z}_{m,i}$ to image $\hat{\mathbf{X}}'_{m,i}$ and feeding it to the target neural network model. Third, the representation $\mathbf{Z}_{m,i}$ is updated by solving an optimization problem to minimize the discrepancy between the dummy gradient $\Delta'^{(k)}_m$ and client gradient $\widetilde{\Delta}^{(k)}_m$. The final reconstruction results are obtained via decoding and postprocessing. Reconstructions can be different based on different postprocessing tools or optimization procedures.

variables, we project the image $\mathbf{X}'_{m,i}$ to a low-dimensional representation $\mathbf{Z}_{m,i}$, which is formulated as

$$\mathbf{Z}_{m,i} = \text{enc}(\mathbf{X}'_{m,i}). \tag{13}$$

The encoding function $\text{enc}(\cdot)$ can be implemented intuitively as a lossy bicubic downsampling function [20], or as complicated as a neural network encoder [5]. In the **second** step, the attacker applies the decoding function $\text{dec}(\cdot)$ to map the representation $\mathbf{Z}_{m,i}$ back to an image $\hat{\mathbf{X}}'_{m,i}$, which is then fed into the federated learning model together with the label $y_{m,i}$ to compute the gradient $\Delta'^{(k)}_m$. **Third**, the attacker solves the following minimization problem (e.g., using Adam [36]) and update the compact representation $\mathbf{Z}_{m,i}$ accordingly:

$$\mathbf{Z}^{\star}_{m,i} = \underset{\mathbf{Z}_{m,i}}{\text{argmin}} \|\Delta'^{(k)}_m - \widetilde{\Delta}^{(k)}_m\|^2. \tag{14}$$

Fourth, after the optimization terminates when the error is below a threshold or after some predefined number of iterations, we decode $\mathbf{Z}^{\star}_{m,i}$ and use postprocessing tools to enhance the image quality. Next, we will present the realizations under the attack framework and make comparisons with existing work.

4.2 Realizations and Comparisons

One implementation of the ROG framework is given as follows. We use independent random variables from uniform distribution $U(0,1)$ to initialize the reconstruction image $\mathbf{X}'_{m,i}$.
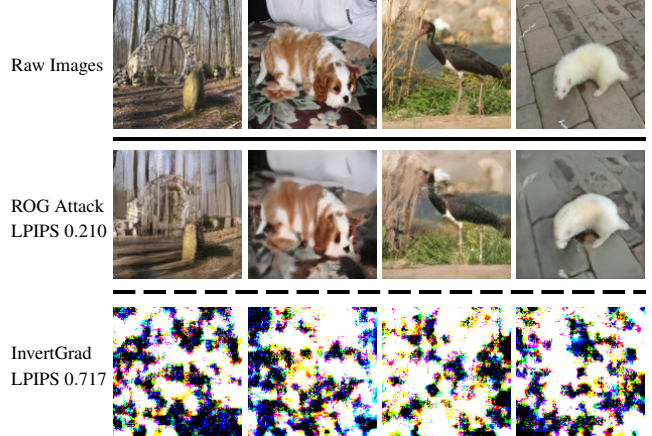

Figure 2: Reconstructed images using two attack schemes on LeNet under FedAvg. ROG reconstruction $\hat{\mathbf{X}}_{m,i}$'s are visually similar to the raw images $\mathbf{X}_{m,i}$'s. In comparison, the attack algorithm InvertGrad fails to recover meaningful private visual information.

To ensure the computational efficiency and avoid vanishing gradient in the backpropagation, we choose bicubic downsampling as the encoding function $\text{enc}(\cdot)$, with a scaling factor of 4. The decoding function $\text{dec}(\cdot)$ corresponds to the bicubic upsampling with the same scaling factor. Finally, we treat the postprocessing in the last step as an image enhancement task. In particular, we randomly apply one of the downsampling approaches, including nearest neighbor, bilinear scaling, and bicubic scaling, and add Gaussian noise to the training examples in the ImageNet dataset [14]. This procedure approximates the effect of the image degradation at different levels [68]. Later, a neural network is optimized with the synthetic dataset to enhance the image quality. More details of the experimental setups and implementations can be found in Appendix A.

Three different model architectures are considered as the instances of shared model in federated learning, namely, LeNet [73], VGG-7 [55], and ResNet-18 [23]. As the batch normalization layers are reported to cause accuracy drop and raise privacy concerns in federated learning [24, 26], we remove these layers in our study and leave them for future work. The attack results that focused on the batch normalization regularizer [65] are not included. For the reconstruction attack, we use the ImageNet validation dataset as the client private data. All images are resized to the resolution of 128×128. We set the batch size to $B = 16$ and the number of local epochs to $\tau = 5$. The learning rate is set to 5×10^{-3}. We keep these settings throughout the paper unless otherwise specified. We randomly select four reconstruction results and compare them with InvertGrad [17]. We report average LPIPS (defined in Appendix B) of the whole batch. Because we observe consistent reconstruction results across three neural network architectures, we present only the LeNet attack results in Figure 2 and

(a)   (b)

Figure 3: Reconstruction results on a whole batch of 16 when raw gradients are known. Compared with (a) the original images, (b) the corresponding ROG reconstructed images are visually similar.

in subsequent experiments. Figure 3 shows the reconstructed images in the full batch, and we compare the best and the worst reconstructed images in Appendix C. The reconstruction on other neural network architectures can be found in Appendix D. ROG reconstruction results are visually similar to the original image $\mathbf{X}_{m,i}$'s. In comparison, InvertGrad fails to reconstruct private image data.

Comparison with InvertGrad. The main differences between ROG and InvertGrad are twofold. First, we propose to find the numerical solution to the optimization problem of (14) in a low dimensional space. Compared to InvertGrad that directly looks for the solution in the original image space, ROG converges faster and yields more stable reconstruction results (see "Ablation study" below). Second, InvertGrad employs the total variation loss $\mathcal{L}_{\text{TV}}$ as an image prior. In our study, we train a neural network model on an additional public image dataset as a postprocessing module, which can be considered as a stronger image prior compared to the total variation regularization term $\mathcal{L}_{\text{TV}}$. Note that InvertGrad fails to reconstruct the data based on the raw gradients, and we do not include their attack results in the subsequent empirical studies. Please refer to Appendix D for the detailed comparison with InvertGrad/DLG on a whole batch of 16.

Ablation Study. We further investigate the effect of two components in ROG attack, including low-dimensional representation and image postprocessing. **First**, we remove the postprocessing module and compare the convergence rate under different dimensions of representations $\mathbf{Z}_{m,i}$. We optimize three autoencoders [5] to obtain different encoding functions $\text{enc}(\cdot)$ and the decoding functions $\text{dec}(\cdot)$ in ROG attack. For autoencoders, we set the dimension d_z of the vectorization of the representation $\mathbf{Z}_{m,i}$ to 8192, 2048, and 512, respectively. In the reconstruction attack, the dimension d_z is equal to the number of unknown variables that an adversary wants to reconstruct. A smaller d_z makes it easier for the attacker to solve the system, but will also lead to more
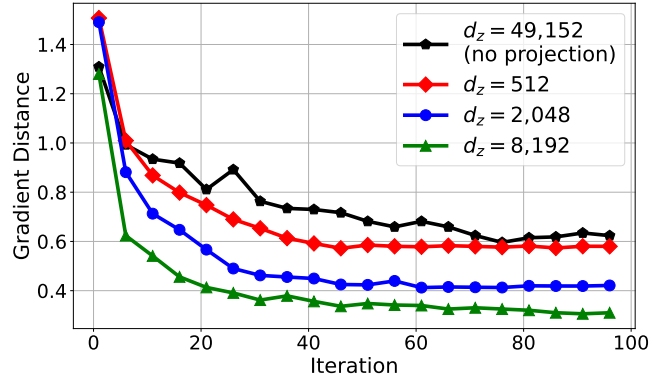

Figure 4: Gradient distance versus the optimization iteration when changing the dimension d_z of the vectorization of the latent representation $\mathbf{Z}_{m,i}$. The three schemes based on latent weight projection converge faster than the baseline "No Projection", which can be treated as using identity functions for $\text{enc}(\cdot)$ and $\text{dec}(\cdot)$. When d_z further decreases, the convergence becomes slower.

information loss during the encoding and decoding procedure. For the baseline method without projection, the number of unknown variables is equal to the number of pixels in the images. In our study, where each RGB image has a resolution of 128×128, the number of unknowns is 49,152. The "no projection" baseline sets encoding function $\text{enc}(\cdot)$ and decoding function $\text{dec}(\cdot)$ to identity functions. The optimization curves of different attack schemes are shown in Figure 4. It can be observed that the three methods with the autoencoder projection converge faster than the baseline method without projection. The results confirm the effectiveness of the latent vector projection in ROG. When the dimension d_z further decreases, the convergence is slower. We visualize the reconstruction when using bicubic downsampled representation ($d_z = 3072$) and autoencoder latent representation ($d_z = 8192$) in Figure 5(a). The number of iterations is set to 100. The autoencoder representation ($d_z = 8192$) gives slightly better results. Overall, reconstructed images are distorted and blurry with certain structural information revealed. **Second**, we remove the low-dimensional projection step and apply the postprocessing module directly. This is equivalent to further postprocess the output of InvertGrad/DLG. The number of iterations is set to 20k, following [17, 65]. The results are shown in Figure 5(b), which are unrecognizable (raw images are in Figure 2). Combing the two components, ROG improves efficiency and reconstruction quality. In particular, the number of iterations can be reduced from 10^4 to 10^2. The quality improvement is more than 70%, improving LPIPS from 0.7 to 0.2. From the two experiments above, we conclude that the low-dimensional representation and postprocessing are both necessary to ensure fast and high-quality reconstruction.

$d_z = 3072$
LPIPS 0.625

$d_z = 8192$
LPIPS 0.502

(a)

Cosine
Distance Loss
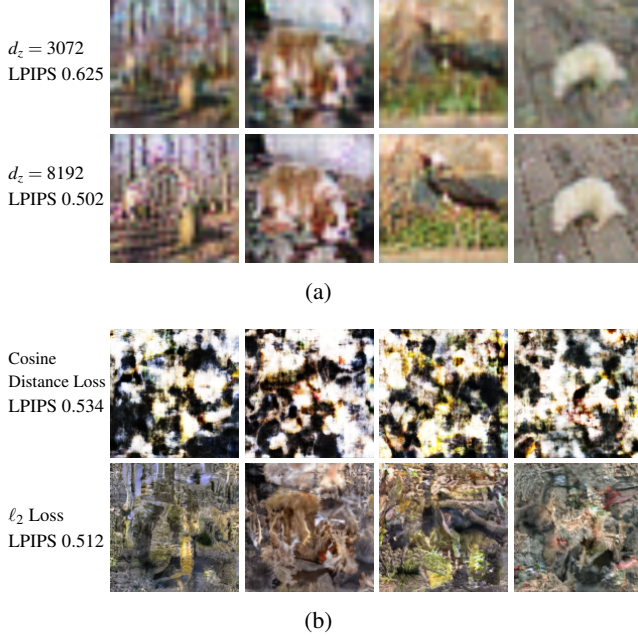LPIPS 0.534

ℓ_2 Loss
LPIPS 0.512

(b)

Figure 5: Ablation study of (a) reconstruction without postprocessing (100 iterations) and (b) reconstruction without using low-dimensional representation (20k iterations). Two modules work synergistically and do not perform well individually.

Raw Images

3-bit ϕ_q
LPIPS 0.213

3-bit ϕ_{qsgd}
LPIPS 0.215

Top-k (0.95)
LPIPS 0.237

Figure 6: Reconstructed images when gradient compression is applied in FedAvg. The compression methods include: 3-bit uniform quantizer ϕ_q, 3-bit QSGD quantizer ϕ_{qsgd}, and Top-k method ϕ_{topk} with the sparsity set to 0.95. The reconstructed images are visually similar to the raw private images.

4.3 Attack on Compressed Gradients

We now discuss the attack results when the compressed gradients are transmitted. The experimental setups are the same as Section 4.2. We use a 3-bit uniform quantizer ϕ_q, a 3-bit QSGD quantizer ϕ_{qsgd}, and the Top-k method ϕ_{topk} with the sparsity parameter equal to 0.95. The reconstruction results are shown in Figure 6. It can be observed that our reconstruction has revealed visual information of the original image data, indicating that even the low precision quantization and sparsification schemes do not provide privacy enhancement despite the information loss. Contrary to prior empirical studies [61, 73], we demonstrate that gradient compression should not be automatically treated as a defense scheme against reconstruction attacks. Prior work on communication-efficient federated learning showed that gradient compression does not significantly affect the model accuracy [3, 4, 51]. From the perspective of the ROG reconstruction, we confirm the intuition that compressed gradients still preserve most of the information with respect to the raw data. As gradient compression does not incorporate privacy protection by design, the observations of failed reconstruction in the literature may give a false sense of security.

Reconstruction from signSGD [8]. The ROG attack method can be extended to the scenario when the 1-bit compressor sign($\cdot$) is adopted. Bernstein et al. [8] showed that the signSGD algorithm converges fast even though only the signs

of the gradients are transmitted. In our attack, we modify the original optimization problem (14) to

$$\mathbf{Z}_{m,i}^{\star} = \underset{\mathbf{Z}_{m,i}}{\operatorname{argmin}} \left\| \tanh(\mathbf{g}') - \operatorname{sign}(\mathbf{g}) \right\|^2, \tag{15}$$

where $\mathbf{g}$ is the client gradient, $\mathbf{g}'$ is the dummy gradient generated in the attack framework, and $\tanh(\cdot)$ is the hyperbolic tangent function approximating the sign($\cdot$) operator. We use the differentiable $\tanh(\cdot)$ function to facilitate the SGD-based optimization of (15). After the optimization terminates, we normalize each image with the maximum absolute values across all coordinates. The results are shown in Figure 7. Since the sign($\cdot$) operator does not preserve the magnitude information, the original normalized images on the second row tend to have a reduced contrast. We further apply the histogram equalization algorithm to enhance the contrast [20]. Counterintuitively, although the 1-bit sign($\cdot$) operator is applied and a great amount of information is discarded, the reconstructed images reveal important information such as the dominant objects in the foreground and the structure information in the background.

5 Attack on Existing Defenses

In this section, we discuss the attack results under three state-of-the-art defense schemes, including Soteria [56], PRE-CODE [54], and FedCDP [62]. All defenses are reported to be effective against the existing gradient leakage attacks, including DLG and InvertGrad. For clarity, we first briefly review these schemes and then provide the empirical attack results.

Raw Images

signSGD
LPIPS 0.347

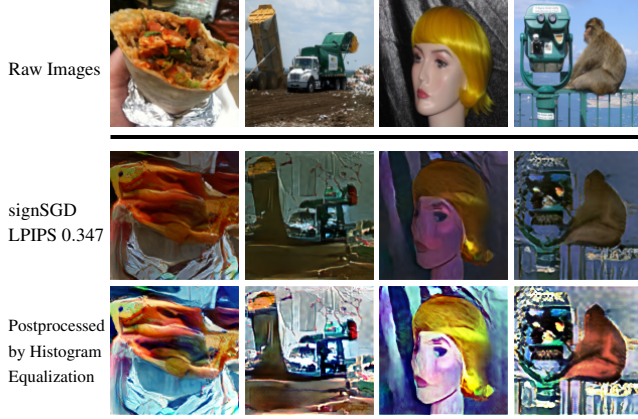Postprocessed
by Histogram
Equalization

Figure 7: Reconstructed images from signSGD. Despite the loss of magnitude information caused by 1-bit quantization, ROG can reconstruct results visually similar to the raw images.

5.1 Review of Defense Schemes

Soteria [56]. Soteria chooses a fully connected layer of the convolutional neural network model as a *defended layer* and perturbs the data representation. Let $\mathbf{X}$ and $\mathbf{X}'$ denote the raw image and reconstructed image via the perturbed representation, respectively. Their corresponding data representations in the defended layer are denoted as $\mathbf{r}$ and $\mathbf{r}'$. To reduce the risk of information leakage, Soteria formulates the problem by maximizing the distance between the original image $\mathbf{X}$ and the reconstructed version $\mathbf{X}'$, while maintaining the similarity between the original representation $\mathbf{r}$ and the target representation $\mathbf{r}'$. The constrained optimization problem is given as:

$$\max_{\mathbf{r}'} \left\| \mathbf{X} - \mathbf{X}' \right\|_2, \tag{16a}$$

$$\text{s.t. } \left\| \mathbf{r} - \mathbf{r}' \right\|_0 \leqslant \rho. \tag{16b}$$

In Soteria, the ℓ_0 norm can ensure the sparsity and its effect can be equivalently represented by a pruning rate. Since this defense has an optimization stage, it will introduce non-negligible computational overhead, especially for the nodes in the fully-connected layers. It has been shown that Soteria is robust and does not impede the convergence of FedAvg [56]. However, this defense does not provide provable privacy protection. For example, Balunović [6] discussed that Soteria only introduced the randomness to the defended layer, and an attacker may still compromise the model by circumventing it. In our attack, we directly reconstruct the noisy image $\mathbf{X}'$ by using the perturbed representation and utilizing the postprocessing tools to enhance the image quality. We will demonstrate that the postprocessing network in the ROG framework is robust to the perturbation in the image space.

PRECODE [54]. PRECODE perturbs the data representation by adding a variational block between two successive layers in a neural network. The variational block is composed of an encoder and a decoder, which are realized as fully connected layers. Given an input $\mathbf{x}$, the encoder projects $\mathbf{x}$ to a representation $\mathbf{z}$, and then generates a feature $\mathbf{b} \sim q(\mathbf{b} \mid \mathbf{z})$ from a Gaussian distribution. The decoder reconstructs a representation $\hat{\mathbf{z}} = p(\mathbf{z} \mid \mathbf{b})p(\mathbf{b})$.

The randomness comes from the sampling of the random vector in the variational block and will affect the gradients in other layers in the backpropagation. Compared to Soteria, an attacker cannot bypass this layer to remove the effect of perturbation. On the other hand, the variational block is available to the attacker based on a white-box model assumption. We let the attacker match the gradients in the variational block without modifying the ROG pipeline. We will show that PRECODE does not effectively protect federated learning against the data reconstruction attack.

FedCDP [62]. FedCDP is a gradient leakage resilient defense based on client level differential privacy (DP) mechanism. It applies per-example gradient clipping and noise injection to achieve provable DP guarantees. In each layer ℓ, the gradient $\mathbf{g}_{m,i,\ell}$ is computed for every individual training example $\mathbf{X}_{m,i}$, and the clipping step is formulated as

$$\hat{\mathbf{g}}_{m,i,\ell} = \mathbf{g}_{m,i,\ell} / \max\left(1, \frac{\|\mathbf{g}_{m,i,\ell}\|_2}{C} \right), \tag{17}$$

where C is a constant of the clipping upper bound. The Gaussian noise vector is then added to the clipped gradient, namely,

$$\phi_{\text{dp}}(\hat{\mathbf{g}}) = \hat{\mathbf{g}} + \mathbf{n}, \quad n_i \sim \mathcal{N}(0, \sigma^2 C^2). \tag{18}$$

Wei et al. [62] have shown that FedCDP can achieve client level per-example differential privacy, and the scheme is robust to data reconstruction attacks.

However, differential privacy may not explicitly protect attribute privacy. For example, Zhang et al. [71] studied the model inversion problem in centralized learning. They found that the attack accuracy remains unchanged despite the various settings of differential privacy budgets. In this work, we empirically study the attack against the client differentially private algorithm in federated learning. Our results indicate that the popular differentially private gradient descent training [1] and the federated implementation [62] may need to be carefully redesigned.

5.2 Attack Results

We first perform the ROG attack against the three defense schemes, Soteria, PRECODE, and FedCDP. We use the same experimental setups as in Section 4.2. For Soteria, we follow Sun et al. [56] and use the configuration of the original paper to achieve a pruning rate of 80% that provides the highest protection level. For FedCDP, we use a clipping upper bound $C = 4$ adopted by Wei et al. [62]. We treat the clipped gradient $\hat{\mathbf{g}}$ as the raw signal and use the signal-to-noise ratio (SNR)

Figure 8: Attack against defense schemes Soteria [56], PRE-CODE [54], and FedCDP [62]. The two feature perturbation based schemes, Soteria and PRECODE, are not effective to defend against the ROG reconstruction attack. Furthermore, FedCDP may not protect attribute privacy effectively. It can be observed that most of the structural information can be reconstructed.

as the noise strength measurement. The results are shown in Figure 8. We set the SNR to 0 dB in FedCDP. The effect of the noise strength is discussed in the next experiment. We observe that the data representation based defenses, including Soteria and PRECODE, can still be vulnerable to the data reconstruction attack. Despite artifacts introduced in the post-processing stage, the reconstructed results are close to the original images. In the meantime, FedCDP does not protect privacy effectively as the structural information and semantics can be revealed. We provide more discussions as follows.

Trade-off between accuracy and privacy. In this experiment, we examine the trade-off between the model test accuracy and privacy leakage. We choose the SNR values from $\{-20, -15, -10, -5, 0\}$ dB for FedCDP, corresponds to $\varepsilon \in \{108.0, 15.2, 4.6, 1.9, 1.0\}$. To quantify privacy leakage, we perform the ROG attack targeting the ImageNet validation dataset assigned to clients. To measure the model accuracy, we choose the CIFAR-10 dataset [37] with $c = 10$ categories as the image classification task in federated learning. We use a different dataset as the original ImageNet training data has been exploited during the postprocessing network training. Following Hsu et al. [25], we simulate the non-IID data with the symmetric Dirichlet distribution [21]. Specifically, for the mth client, we draw a random vector $q_m \sim \text{Dir}(\alpha)$, where $q_m = [q_{m,1}, \ldots, q_{m,c}]^\top$ belongs to the $(c-1)$-standard simplex. Images with category k are assigned to the mth client in proportional to $(100 \cdot q_{m,k})\%$. The parameter α is set to

0.5. A total of 100 clients are considered, and 10 of them are selected with equal probability in each communication round. We terminate the training after 100 communication rounds and use the model accuracy as the utility metric. The trade-off between the model accuracy and privacy leakage of different methods is shown in Figure 9(a). In general, defense schemes sacrifice the model accuracy to reduce privacy leakage. The cross points in the upper right corner represent desired defense schemes that achieve high model accuracy and privacy protection. The proposed ROG attack shifts the achievable utility–privacy region to the bottom left, indicating that existing privacy-preserving federated learning algorithms may give a false sense of security. FedAvg and the three defense schemes can be effective against the InvertGrad attack, which may be misleadingly classified as desirable defense schemes. We present qualitative reconstruction results under different FedCDP settings in Figure 9(b). When the SNR decreases, the reconstructed results gradually degrade. Even when the SNR is low, some structural information can still be revealed. We note that there exists a relatively smooth transition for the attack results against different defense settings. As a result, it is ambiguous to tell whether an attack is successful or not.

Based on the observation, we point out that existing federated learning algorithms and defenses can still be vulnerable to reconstruction attacks. Differential privacy tools may not explicitly protect data attributes, such as the pixels of images in our simulation, while they could sacrifice the model utility to a large extent. Meanwhile, other defense schemes that do not rely on the differential privacy concept were evaluated on some specific attack designs without provable privacy guarantees. We have shown that these heuristic defenses do not provide satisfactory privacy protection. Our results indicate the importance of redesigning the privacy-preserving framework for federated learning in parallel with the existing differential privacy paradigm.

Unevaluated defenses. There exist other defense schemes that we do not evaluate in this paper. For example, Huang et al. [26, 27] proposed to encode the images by compositing several training examples and flipping the signs of the data randomly. The vulnerability of data encoding has been shown in recent work [12]: with a well-designed attack that formulated the problem as a noisy linear system, the original images can be recovered. Other defenses combine differential privacy with secure aggregation [2, 33] to achieve stronger protection. Some studies have shown that the individual gradient can be recovered in secure aggregation [39]. A comprehensive evaluation of these defenses is out of the scope of this paper. The design and evaluation of hybrid defenses are active research fields and we leave them for future work.
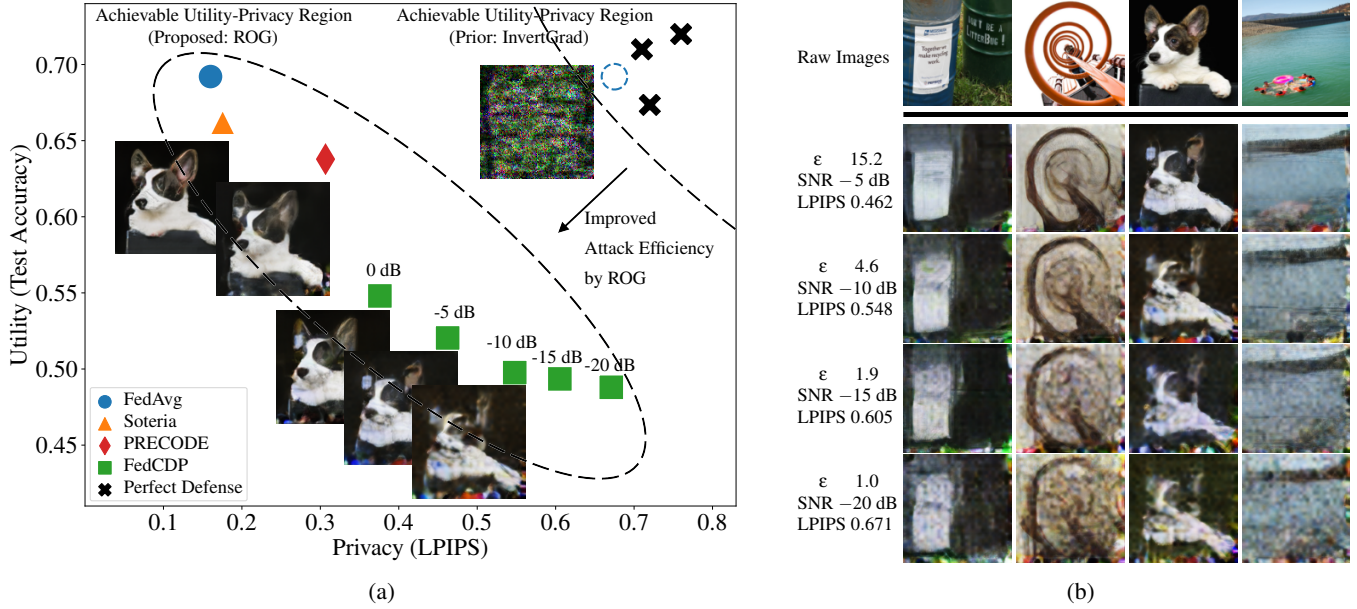
(a)

(b)

Figure 9: (a) Trade-off between model accuracy and privacy under the ROG attack. A larger LPIPS value indicates better privacy protection against the attack. The cross points in the upper-right corner represent the desired/unrealistic defense schemes achieving high model accuracy and privacy protection. The dashed circle denotes the operating point of FedAvg under InvertGrad attack, which is misleadingly classified to the same region as the desired defense schemes and gives a false sense of security. Meanwhile, the three defense schemes have been verified to be effective against the InvertGrad attack, which may also be classified as desired defense schemes. The achievable utility–privacy region is shifted to the bottom left due to the proposed ROG attack. (b) ROG attack results when FedCDP is used as the defense scheme. The reconstructed images gradually degrade as the defense strength increases (or the SNR decreases). When the SNR is -20 dB, some structural information can still be revealed whereas the accuracy has dropped severely compared to FedAvg.

6 Attack at a Semantic Level

In this section, we present a variant of the ROG attack by using a different postprocessing module. We propose to reconstruct a good-quality image at a semantic level (ROGS). Instead of focusing on the pixel-level error between the original image and the reconstructed counterpart, ROGS leverages a pretrained generative model G, such as BigGAN [11], to synthesize realistic images while maintaining the semantics of the original image. The generative model G maps a latent vector $\mathbf{z}$ to the image space, and the output has a similar distribution to the real images. We search the latent space of the generative model G and restrict the similarity between the generated image and the original ROG lower-quality reconstruction. The goal of ROGS attack is to reconstruct an image sharing the same knowledge and semantics as in the original private image. Image reconstruction via the ROGS attack differs from the notion of property inference attack [44] that target sensitive information. Sensitive information is generally difficult to define and corresponding privacy relies on personal preference [48]. Compared to property inference, such as recovering gender, age, or race, ROGS may be viewed as a more general attack that diversifies the concept of com-

promising privacy. It has the capability to reveal privacy even if some sensitive information is not present in the corpus.

From the technical perspective, we let ROGS invert an input to the GAN latent space, which is also known as the GAN inversion task in the literature [30,64]. The optimization problem can be formulated as

$$\mathbf{z}_{m,i}^{\star} = \operatorname*{argmin}_{\mathbf{z}_{m,i}} \mathcal{L}(G(\mathbf{z}_{m,i}), \hat{\mathbf{X}}_{m,i}), \qquad (19)$$

where $\mathcal{L}$ is a loss function quantifying the distance between two images, $\mathbf{z}$ denotes a latent vector, and $\hat{\mathbf{X}}$ is the original ROG output. The ROGS attack will take the latent vector $\mathbf{z}_{m,i}^{\star}$ as the input and output $G(\mathbf{z}_{m,i}^{\star})$ as the attack result.

In our implementation, we use a combination of the ℓ_2 distance and the LPIPS metric, namely,

$$\mathcal{L}(\mathbf{X}_1, \mathbf{X}_2) = \lambda_1 \cdot \|\mathbf{X}_1 - \mathbf{X}_2\|_2 + \lambda_2 \cdot \mathrm{LPIPS}(\mathbf{X}_1, \mathbf{X}_2), \quad (20)$$

where λ_1 and λ_2 are the coefficients balancing the two terms. The loss function encourages the reconstructed image to preserve the structure with the ℓ_2 distance and the perceptual styles of the original image with the LPIPS score. We choose the BigGAN model [11] as the generative model G, which is a class conditional GAN model trained on the ImageNet

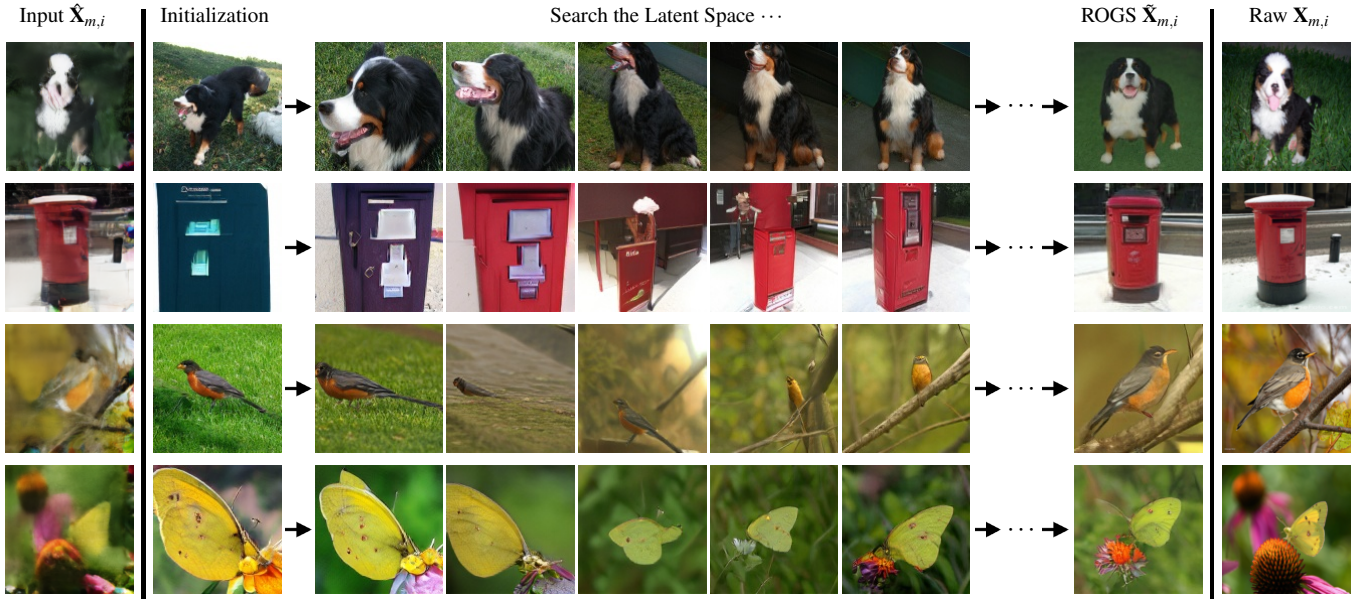| Input $\hat{\mathbf{X}}_{m,i}$ | Initialization | Search the Latent Space $\cdots$ | | | | | ROGS $\tilde{\mathbf{X}}_{m,i}$ | Raw $\mathbf{X}_{m,i}$ |

Figure 10: Reconstruction attack at a semantic level using ROGS. Given lower-quality inputs $\hat{\mathbf{X}}_{m,i}$ from ROG, the optimization starts from a random latent vector corresponding to an image within the same class category as the ROG output. The foreground and background will gradually change to match the lower-quality input during the latent-space search. With some compromise at the pixel-level, the final results $\tilde{\mathbf{X}}_{m,i}$ can better reveal the semantic visual information of the original images.

dataset. In particular, given a latent vector $\mathbf{z}_{m,i}$ and a class label $y_{m,i}$, the output of the model is constrained to be within the specific class. We start from a random initialization in the latent space and use the Adam optimizer to solve (19).

We use lower-quality images obtained in the ROG attack and show the attack results of ROGS in Figure 10. It can be observed that the optimization process begins with a random initialization from the same image class, and gradually changes the foreground and background during the latent space search. In the first row, the raw image shows that a Bernese mountain dog is standing on the grass, facing the camera in the dark. The initialization chooses a dog of the same breed running on the grass in the daytime. During the optimization, the dog's posture and orientation are changed smoothly to match the lower-quality input. Meanwhile, the background is also gradually changed to the grass in the dark night. Although the reconstruction is not pixel-wise accurate, privacy leakage can still happen when rich semantics are revealed. Likewise, the details of the mailbox on the second row, including the shape, color, and layout, are adapted to match the counterpart in the input. The snow and the road in the background are reconstructed after the optimization. In the third row, the robin's size, coat color, and posture have been changed to a similar style as the original image after the latent space search, as well as the seasonal information. Similar changes can be observed for the sulphur butterfly picture on the fourth row, where the orientation and size of the butterfly are adjusted during the optimization.

We now discuss how to quantify privacy leakage at the

semantic level. We start by raising two concerns on current reconstruction evaluation metrics adopted in the literature based on the reconstruction results in Sections 4–5. First, we have observed that the attack success rate suggested by Wei et al. [61] may not be a good indicator in some scenarios. For the reconstruction under existing defense schemes, privacy leakage should not be treated as a binary quantity. Merely using a threshold may not be a good indicator for a successful attack. The structural information or semantics can be revealed in the reconstructed image, although the results do not perfectly match the raw images. There exists ambiguity in telling whether the attack is successful or not, let alone using an algorithm to automatically calculate the attack success rate. Second, the pixel-level based metrics, including MSE, PSNR, and SSIM, may fail to capture privacy leakage in some scenarios. For example, for the attack against signSGD in Figure 7, the average SSIM is 0.37, which may not be able to fully capture the perceptual similarity due to the color jitter effect. A low PSNR value or a high MSE may give misleading conclusions on privacy leakage.

To measure privacy leakage at a semantic level, we use a state-of-the-art multi-label classification network [7] to tag the images. Specifically, the classification network has been trained on OpenImage (V6) [38], which contains 9,600 classes, including the object category, color, season, etc. Given two images $\mathbf{X}_1$ and $\mathbf{X}_2$, suppose their detected tags are denoted by set A and B, respectively. The Jaccard similarity [29]

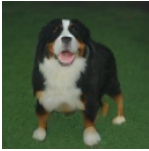Table 1: Image pair and corresponding tags in the ROGS attack.

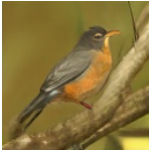| Image	Tags	Image	Tags
	Black, Dog, White, Brown, Green, Snout, Grass, Tints and shades, Turquoise (Color), Habitat, Beige, Grey, Carnivore, Bernese mountain dog, Light, Limb, ...		Post box, Mailbox, Red, Daytime, Maroon, Infrastructure, Public space, White, Snow, Carmine, Line, Amber (Color), Morning, Circle, Material, ...
	Dog, Snout, Black, Brown, White, Bernese mountain dog, Green, Tints and shades, Habitat, Grass, Carnivore, Shoe, Nature, Turquoise (Color), ...		Post box, Red, Mailbox, Snow, Daytime, White, Maroon, Carmine, Public space, Composite material, Structure, Line, Product, Winter, ...
	Bird, Brown, Beak, Orange (Color), Feather, Habitat, Daytime, Amber (Color), American robin, Morning, Branch, Twig, Ivory (Color), ...		Pollinator, Moths and butterflies, Flowering plant, Flower, Arthropod, Insect, Yellow, Butterfly, Magenta, Green, Purple coneflower, Spring (Season), ...
	Bird, Beak, Brown, Feather, Habitat, Orange (Color), Amber (Color), Daytime, Ivory (Color), Ecoregion, Blond, Orange (Color), Peach (Color), ...		Pollinator, Arthropod, Moths and butterflies, Butterfly, Insect, Invertebrate, Yellow, Wing, Flowering plant, Green, Ecosystem, Computer wallpaper, ...

Table 2: Comparisons between ROG and ROGS with different metrics. ROGS improves the semantic similarity score.

Image	Attack	PSNR	SSIM	LPIPS	Jaccard
Dog	ROG	22.1 dB	0.897	0.332	0.400
	ROGS	14.9 dB	0.611	0.348	0.632
Mailbox	ROG	21.3 dB	0.946	0.206	0.433
	ROGS	14.5 dB	0.832	0.237	0.600
Robin	ROG	20.8 dB	0.836	0.299	0.250
	ROGS	15.3 dB	0.285	0.350	0.600
Butterfly	ROG	21.3 dB	0.900	0.287	0.312
	ROGS	13.9 dB	0.334	0.458	0.579
Random Noise[1]		6.5 dB	0.006	1.409	0.030
Same Class Image[2]		7.7 dB	0.070	0.631	0.248

[1] Average similarity between ROGS and random Gaussian noise over five repetitions.
[2] Average similarity between ROGS and images from the same ImageNet class over five repetitions.

between A and B is given as follows:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}, \qquad (21)$$

where $|\cdot|$ denotes the cardinality of the set. The Jaccard similarity ranges from 0 to 1 by design.

The evaluation results of the same set of images in Figure 10 are given in Table 1 and Table 2. Table 1 lists the

pairs of the original image and the reconstructed one, along with the detected tags excluding some generic tags such as "Photograph", "World", and "Color". Table 2 gives the reconstruction quality scores. It is observed that the tags of the dominating objects can be detected, including "Bernese mountain dog", "Post box", "Bird", and "Butterfly". The background information is also included in the tags, such as "Grass", "Snow", "Twig", and "Purple coneflower". The average PSNR, SSIM, and LPIPS scores compared to an image randomly selected from the same class are around 7.7 dB, 0.07, 0.63, respectively. PSNR values of ROGS results are around 14 dB, which are worse than the ROG reconstruction and better than the random cases. In addition, ROGS has slightly worse LPIPS values and worse SSIM compared to ROG. In other words, ROGS does not directly improve the image similarity at the pixel level as compared to ROG, but consistently outperforms ROG in terms of Jaccard similarity scores, which is more oriented toward recognition at the semantic level. We now compare the difference between the SSIM value and the Jaccard similarity. For the first pair of Bernese mountain dogs, SSIM and Jaccard values are above 0.6 and tend to agree with each other. For the second pair of post boxes, the SSIM gives a much higher value, 0.832, compared to Jaccard similarity, 0.6. A large discrepancy between SSIM and Jaccard can be found in the third and fourth pairs. For the third pair of robins, the SSIM is 0.285, which is relatively low. In the meantime, the semantics have been
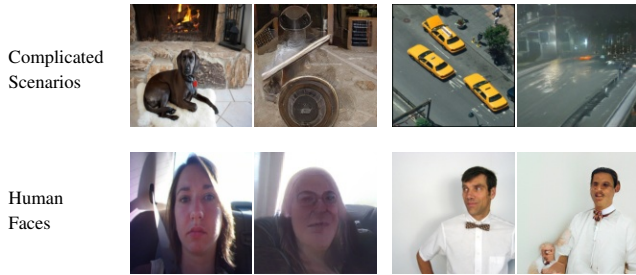
Complicated Scenarios

Human Faces

Figure 11: Image pairs of the raw data and the reconstruction that deviate at the semantic level. On the first row, ROGS fails to reconstruct images containing complicated scenarios. For the second row involving human faces, the reconstruction fails to reveal their identities.

successfully reconstructed, which is also reflected in a Jaccard similarity of 0.6. A similar pattern can be observed in the fourth pair. The reconstruction has revealed important information about butterflies resting on the flower. Such a privacy loss may be ignored when merely checking the SSIM value of 0.334.

Limitations. We would like to clarify that the aforementioned Jaccard similarity is not intended to replace the current metrics for image privacy loss measurement, but rather to serve as a supplement or as a motivation to help us rethink the privacy leakage problem. In the meantime, there still exist some limitations in the reconstruction and evaluation at the semantic level. We give some negative reconstruction examples in Figure 11. When the raw images contain a complicated scenario, such as including multiple dominating objects, the attack algorithm may fail to converge. This can be observed in the first row of Figure 11. Another example is when the raw image involves human faces, the reconstruction may not be able to reveal the person's identity. This may not be considered as a severe privacy leakage in some applications, such as facial data analysis. Furthermore, not all of the semantics can be included in the tags, such as the action, orientation, and body size. In addition, the multi-label classification neural network may not always give the correct tags. For example, a tag "Shoe" is detected in the second row of Table 1, which can be considered as a misclassification. The importance of different tags and their correlations may also need to be considered to improve the similarity score design. A more comprehensive study is out of the scope of this work. However, we hope these results can inspire researchers to revisit the privacy leakage issues in federated learning.

7   Discussion and Future Work

The concept of federated learning was proposed to preserve sensitive client data for multi-party machine learning. In this work, we have shown that it is possible to reconstruct client data with high quality from noisy gradients in a more real-

istic federated learning setting compared to existing attacks. Contrary to prior empirical work, we have demonstrated that gradient compression and perturbation cannot be treated as effective privacy protection strategies. Our attack scheme has also successfully reconstructed private information under existing defenses. Based on these observations, we conjecture that for federated learning algorithms that do not provide privacy guarantees for data attributes, an adversary can always find a certain attack scheme to disclose the privacy in the raw data, as long as the mutual information between gradient and raw data is not close to zero.

In the future, it will be interesting to investigate the trade-off between utility and privacy for hybrid defenses. As it has been observed in [26], a combination of multiple defenses can provide better privacy protection. Besides, differentially private training with secure aggregation [2,33] may also have potential to defend against the proposed ROG attack. In parallel to the existing differential privacy paradigm, a more general formulation of the privacy concept will be worth exploring. We also believe that the study of privacy leakage beyond the image classification task will provide more insights to the research community.

References

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

[2] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpSGD: Communication-efficient and differentially-private distributed SGD. *Advances in Neural Information Processing Systems*, 31, 2018.

[3] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. In *Conference on Empirical Methods in Natural Language Processing*, pages 440–445, 2017.

[4] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

[5] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 37–49. JMLR Workshop and Conference Proceedings, 2012.

[6] Mislav Balunović, Dimitar I Dimitrov, Robin Staab, and Martin Vechev. Bayesian framework for gradient leak-

age. In *International Conference on Learning Representations*, 2022.

[7] Emanuel Ben-Baruch, Tal Ridnik, Itamar Friedman, Avi Ben-Cohen, Nadav Zamir, Asaf Noy, and Lihi Zelnik-Manor. Multi-label classification with partial annotations using class-aware selective loss. *arXiv preprint arXiv:2110.10955*, 2021.

[8] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. SignSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569, 2018.

[9] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *NeurIPS 2016 workshop on Private Multi-Party Machine Learning*, 2016.

[10] Antoine Boutet, Thomas Lebrun, Jan Aalmoes, and Adrien Baud. MixNN: Protection of federated learning against inference attacks by mixing neural network layers. *arXiv preprint arXiv:2109.12550*, 2021.

[11] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[12] Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Abhradeep Thakurta, and Florian Tramèr. Is private learning possible with instance encoding? In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 410–427. IEEE, 2021.

[13] Pau-Chen Cheng, Kevin Eykholt, Zhongshu Gu, Hani Jamjoom, KR Jayaram, Enriquillo Valdez, and Ashish Verma. Separation of powers in federated learning. *arXiv preprint arXiv:2105.09400*, 2021.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[15] Jieren Deng, Yijue Wang, Ji Li, Chenghong Wang, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. Tag: Gradient attack on transformer-based language models. In *Findings of the Association for Computational Linguistics*, pages 3600–3610, 2021.

[16] Liam Fowl, Jonas Geiping, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models. *arXiv preprint arXiv:2110.13057*, 2021.

[17] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*, 2020.

[18] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, 2016.

[19] Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, 2021.

[20] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing, 3rd Edition*. 2014.

[21] Irving J Good. On the application of symmetric Dirichlet distributions and their mixtures to contingency tables. *The Annals of Statistics*, 4(6):1159–1189, 1976.

[22] Hanlin Gu, Lixin Fan, Bowen Li, Yan Kang, Yuan Yao, and Qiang Yang. Federated deep learning with Bayesian privacy. *arXiv preprint arXiv:2109.13012*, 2021.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398, 2020.

[25] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[26] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[27] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. In *International conference on machine learning*, pages 4507–4518. PMLR, 2020.

[28] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[29] Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New Phytologist*, 11(2):37–50, 1912.

[30] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2019.

[31] Jinwoo Jeon, Kangwook Lee, Sewoong Oh, Jungseul Ok, et al. Gradient inversion with generative image prior. *Advances in Neural Information Processing Systems*, 34:29898–29908, 2021.

[32] Malhar S Jere, Tyler Farnan, and Farinaz Koushanfar. A taxonomy of attacks on federated learning. *IEEE Security & Privacy*, 19(2):20–28, 2020.

[33] Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*, pages 5201–5212. PMLR, 2021.

[34] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1), 2021.

[35] Raouf Kerkouche, Gergely Ács, Claude Castelluccia, and Pierre Genevès. Compression boosts differentially private federated learning. In *European Symposium on Security and Privacy*, pages 1–15, 2021.

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.

[37] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master thesis, Dept. of Comput. Sci., Univ. of Toronto, Toronto, Canada*, 2009.

[38] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[39] Maximilian Lam, Gu-Yeon Wei, David Brooks, Vijay Janapa Reddi, and Michael Mitzenmacher. Gradient disaggregation: Breaking privacy in federated learning by reconstructing the user participant matrix. In *International Conference on Machine Learning*, pages 5959–5968. PMLR, 2021.

[40] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[41] Yong Liu, Xinghua Zhu, Jianzong Wang, and Jing Xiao. A quantitative metric for privacy leakage in federated learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

[42] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.

[43] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 2017.

[44] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019.

[45] Fan Mo, Anastasia Borovykh, Mohammad Malekzadeh, Hamed Haddadi, and Soteris Demetriou. Quantifying and localizing private information leakage from neural network gradients. *arXiv preprint arXiv:2105.13929*, 2021.

[46] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019.

[47] Dinh C Nguyen, Ming Ding, Quoc-Viet Pham, Pubudu N Pathirana, Long Bao Le, Aruna Seneviratne, Jun Li, Dusit Niyato, and H Vincent Poor. Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet of Things Journal*, 2021.

[48] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Towards a visual privacy advisor: Understanding and predicting privacy risks in images. In *Proceedings of the IEEE international conference on computer vision*, pages 3686–3695, 2017.

[49] Xudong Pan, Mi Zhang, Yifan Yan, Jiaming Zhu, and Min Yang. Theory-oriented deep leakage from gradients via linear equation solver. *arXiv preprint arXiv:2010.13356*, 2020.

[50] Jia Qian, Hiba Nassar, and Lars Kai Hansen. Minimal conditions analysis of gradient-based reconstruction in federated learning. *arXiv preprint arXiv:2010.15718*, 2020.

[51] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[52] Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *arXiv preprint arXiv:2007.07646*, 2020.

[53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[54] Daniel Scheliga, Patrick Mäder, and Marco Seeland. PRECODE – a generic model extension to prevent deep gradient leakage. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1849–1858, 2022.

[55] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[56] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9311–9319, 2021.

[57] Aidmar Wainakh, Till Müßig, Tim Grube, and Max Mühlhäuser. Label leakage from gradients in distributed machine learning. In *Annual Consumer Communications & Networking Conference*, 2021.

[58] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H. Brendan McMahan, Blaise Aguera y Arcas, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

[59] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[60] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Hang Su, Bo Zhang, and H. Vincent Poor. User-level privacy-preserving federated learning: Analysis and performance optimization. *IEEE Transactions on Mobile Computing*, 2021.

[61] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating client privacy leakages in federated learning. In *Computer Security – ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I*, page 545–566, Berlin, Heidelberg, 2020. Springer-Verlag.

[62] Wenqi Wei, Ling Liu, Yanzhao Wut, Gong Su, and Arun Iyengar. Gradient-leakage resilient federated learning. In *International Conference on Distributed Computing Systems*, 2021.

[63] Yuxin Wen, Jonas Geiping, Liam Fowl, Micah Goldblum, and Tom Goldstein. Fishing for user data in large-batch federated learning via gradient magnification. In *International Conference on Machine Learning*, pages 23668–23684. PMLR, 2022.

[64] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN inversion: A survey. *arXiv preprint arXiv:2101.05278*, 2021.

[65] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021.

[66] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations*, 2021.

[67] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning. In *USENIX Annual Technical Conference*, pages 493–506, 2020.

[68] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *IEEE/CVF International Conference on Computer Vision*, pages 4791–4800, 2021.

[69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[70] Rui Zhang, Song Guo, Junxiao Wang, Xin Xie, and Dacheng Tao. A survey on gradient inversion: Attacks, defenses and future directions. *arXiv preprint arXiv:2206.07284*, 2022.

[71] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 253–261, 2020.

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

[73] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, 2019.

A Implementation Details

We provide more details of the postprocessing neural networks as follows. We adopt the U-Net architecture [53] as the GAN generator, which is composed of an encoder that downsamples the images and a decoder that upsamples the feature maps back to the original size. This architecture has been widely adopted in image-to-image translation [28] and image inpainting [46]. In the generator, we take the form of convolution-InstanceNorm-ReLu for the backbone module, and downsample the input image four times. We add skip connections between the encoder and the decoder to circumvent severe information loss caused by the downsampling. The discriminator architecture is based on PatchGAN [46].

To synthesize the dataset, we randomly apply one of the downsampling approaches, including nearest neighbor, bilinear scaling, and bicubic scaling, and add Gaussian noise to the training examples in the ImageNet dataset. In particular, we randomly choose the order of downsampling, noise injection, and a Gaussian blur kernel to degrade the original images. We use a linear combination of the ℓ_1 loss and the adversarial loss as the cost function. The Adam optimizer is adopted for the GAN training, and the learning rate is set to 1×10^{-5} for both the generator and the discriminator. For ROGS, we set $\lambda_1 = \lambda_2 = 1$ and optimize for 1000 iterations. The pretrained models and implementation of the attack are available at https://anonymous.4open.science/r/rog-DC5E.

B Image Quality Metrics

We review the image quality metrics used in this work, including PSNR, SSIM, and LPIPS. Given two images $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{h \times w}$, the mean squared error (MSE) is defined as:

$$\text{MSE}(\mathbf{X}, \mathbf{Y}) = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} (X_{i,j} - Y_{i,j})^2, \qquad (22)$$

where $X_{i,j}$ denotes the (i,j)th entry of the matrix $\mathbf{X}$. The PSNR in dB is given by:

$$\text{PSNR}(\mathbf{X}, \mathbf{Y}) = 10 \cdot \log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}}\right), \qquad (23)$$

where MAX_I is the maximum possible pixel value of the image. By definition, PSNR is a pixel-wise similarity metric. SSIM better approximates the perception model. Given two

(a)   (b)

Figure 12: Reconstruction results a whole batch of 16 when a 3-bit QSGD quantizer ϕ_{qsgd} is applied. Compared with (a) raw images, (b) ROG reconstructed images are visually similar.

images $\mathbf{X}, \mathbf{Y}$, suppose μ_X and μ_Y are the average of $\mathbf{X}$ and $\mathbf{Y}$, σ_X^2 and σ_Y^2 are the variance of the two images, and σ_{XY} is the covariance. The SSIM is given by:

$$\text{SSIM}(\mathbf{X}, \mathbf{Y}) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)}, \qquad (24)$$

where c_1 and c_2 are two variables to stabilize the division. By design, $\text{SSIM} \in [0, 1]$, and a higher value indicates a higher similarity.

Compared to the traditional image quality metrics, LPIPS was proposed based on the activations of convolutional neural networks. Consider the lth layer of a neural network with unit-normalized feature $A^l \in \mathbb{R}^{H_l \times W_l \times C_l}$, where H_l, W_l, and C_l denote its height, width, and channels, respectively. With predefined coefficients $\mathbf{c}_l \in \mathbb{R}^{C_l}$, the LPIPS value can be calculated as

$$\text{LPIPS}(\mathbf{X}, \mathbf{Y}) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|\mathbf{c}_l \odot (A^l_{h,w} - B^l_{h,w})\|_2^2, \qquad (25)$$

where $A^l_{h,w}$ and $B^l_{h,w}$ denote the feature maps of two images.

C More Reconstruction Examples

We provide more reconstruction results on the whole batch in addition to those given in Section 4 and Figure 3. The reconstruction of a batch size of 16 under a 3-bit QSGD quantizer is shown in Figure 12. We compare the best and worst reconstruction results in Figure 13 when different batch sizes are used. The best and worst results are selected based on LPIPS values. In Figure 14, we show the best 16 and worst 16 reconstructed images when the batch size is set to 128. The best reconstructed images convey meaningful pictorial information, whereas the worst reconstructed images are almost unrecognizable.
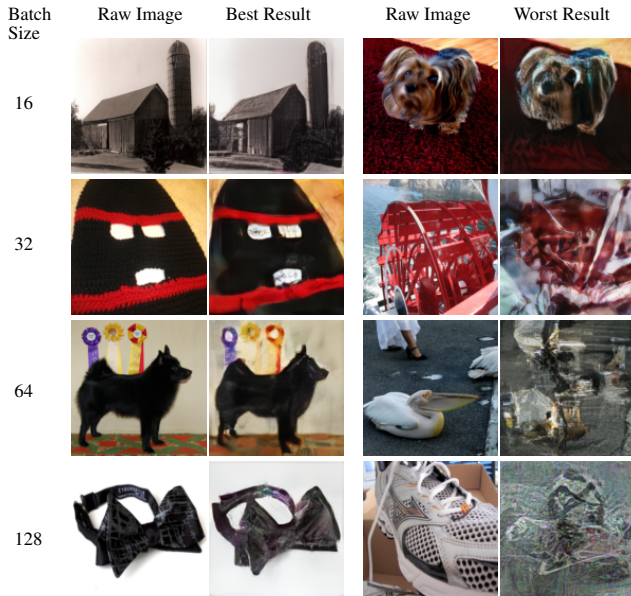
Figure 13: Comparison between the best and the worst reconstructed images in a full batch. The best reconstructed images convey meaningful pictorial information, whereas the worst reconstructed images are almost unrecognizable.

D Additional Experiments

We discuss some additional simulation results in this section. First, we demonstrate the full batch reconstruction with InvertGrad and DLG in Figure 15. For InvertGrad, we use the Adam optimizer and set the number of iterations to 24k. For DLG, we use L-BFGS optimizer and set the number of iterations to 300. We use the original implementations provided by [17, 73].

Neural Network Architectures In this experiment, we study the impact of different neural network architectures on the attack scheme. We choose the LeNet, VGG-7, and ResNet-18 and demonstrate the reconstruction results in Figure 16.
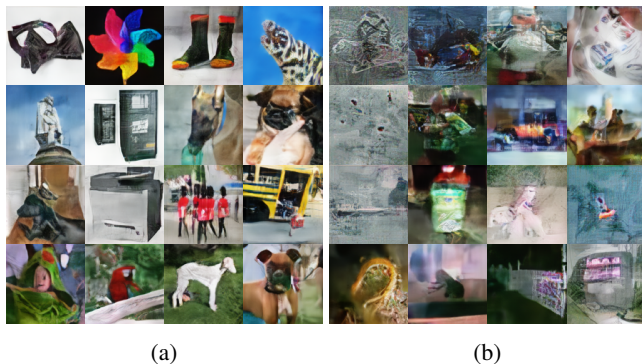

Figure 14: (a) The best 16 reconstructed images and (b) the worst 16 reconstructed images from a full batch of size 128.
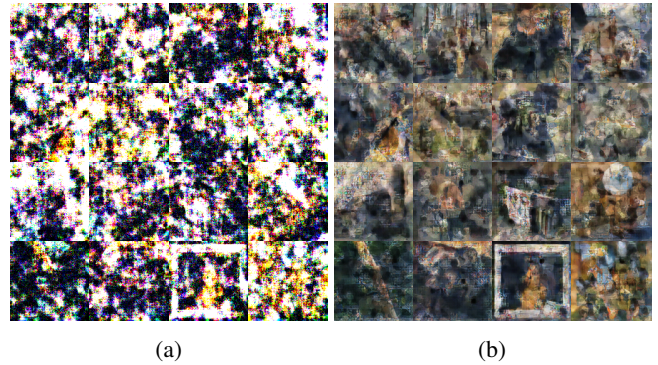

Figure 15: Reconstruction results on a whole batch of 16 with the attack methods (a) InvertGrad with a postprocessing module and (b) DLG with a postprocessing module.
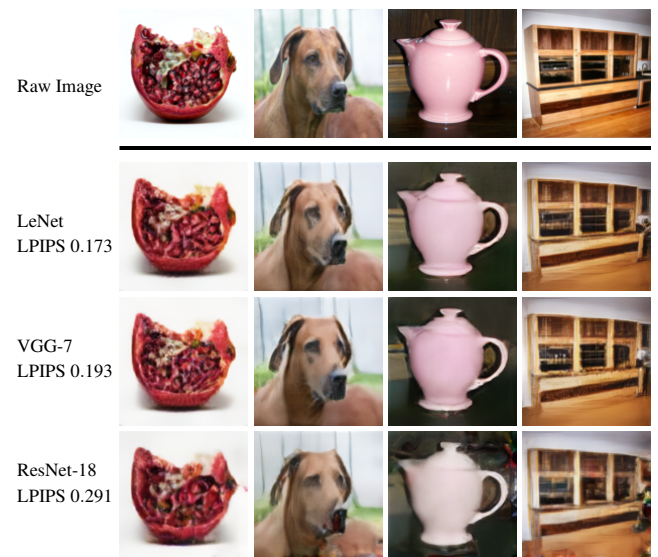

Figure 16: Attack against different neural network architectures. All reconstructed images are visually similar to raw images.

LeNet is a five-layer convolutional network with 5×5 kernels, and VGG-7 is a seven-layer convolutional network with 3×3 kernels. ResNet-18 is an eighteen-layer residual network with skip connections. Intuitively, the increased number of nodes in a neural network appears to give the adversary more advantages, as the number of known conditions will also increase. On the other hand, more sophisticated neural network architecture will have an increased nonlinearity, thus impeding a successful reconstruction. In Figure 16, we can observe that all reconstructed images under different neural network architectures are visually similar. Our observation is consistent with the prior study that changing the neural network architecture may not directly affect the privacy protection level. A more extensive study concerning neural network architectures is warranted in future work.