CGIR: Conditional Generative Instance Reconstruction Attacks against Federated Learning

Xiangrui Xu, Pengrui Liu, Wei Wang, Hong-Liang Ma, Bin Wang, Zhen Han, and Yufei Han

Abstract—Data reconstruction attack has become an emerging privacy threat to Federal Learning (FL), inspiring a rethinking of FL's ability to protect privacy. While existing data reconstruction attacks have shown some effective performance, prior arts rely on different strong assumptions to guide the reconstruction process. In this work, we propose a novel Conditional Generative Instance Reconstruction Attack (CGIR attack) that drops all these assumptions. Specifically, we propose a batch label inference attack in non-IID FL scenarios, where multiple images can share the same labels. Based on the inferred labels, we conduct a "coarse-to-fine" image reconstruction process that provides a stable and effective data reconstruction. In addition, we equip the generator with a label condition restriction so that the contents and the labels of the reconstructed images are consistent. Our extensive evaluation results on two model architectures and five image datasets show that without the auxiliary assumptions, the CGIR attack outperforms the prior arts, even for complex datasets, deep models, and large batch sizes. Furthermore, we evaluate several existing defense methods. The experimental results suggest that pruning gradients can be used as a strategy to mitigate privacy risks in FL if a model tolerates a slight accuracy loss.

Index Terms—Data Reconstruction Attacks, Privacy, Federated Learning.

1 Introduction

With the proliferation of data silos and the heightened awareness of privacy issues, traditional centralized machine learning frameworks are facing efficiency and privacy issues [1]. Federated learning (FL) has recently been proposed as a novel distributed machine learning paradigm, where several clients can jointly train a global model by sharing only the gradients during training [2] [3] [4]. It may appear safe at first glance, but the gradients as a mapping of data in the training model pose a potential privacy risk. For example, an attacker can determine whether a training sample is involved in the training process [5] [6], or determine what properties the training data have [7] [8]. With the right attack, the attacker can even enable a detailed image reconstruction at the pixel level [11] [13] [14]. This will directly result in a privacy breach to the participants. Therefore, analyzing and exploring the privacy vulnerabilities of FL is critical to its efficient development and deployment.

- *Xiangrui Xu, Pengrui Liu, Wei Wang and Zhen Han are with Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing, 100044, China. Wei Wang and Bin Wang are the corresponding authors.*
  E-mail: xiangrui.xu@bjtu.edu.cn, pengrui.liu@bjtu.edu.cn, wangwei1@bjtu.edu.cn, zhan@bjtu.edu.cn.
- *Hong-Liang Ma is with school of information science and technology, Shihezi University, 832000, China.*
  E-mail: mhl_inf@shzu.edu.cn.
- *Bin Wang is with Zhejiang Key Laboratory of Multi-dimensional Perception Technology, Application and Cybersecurity, Hangzhou, 310053, China.*
  E-mail: bin_wang@zju.edu.cn.
- *Yufei Han is with INRIA, 35042, Rennes, Bretagne, France.*
  Email: yufei.han@inria.fr.

- *The code is available at https://github.com/abcerger/CGIR-attack.*

In this work, we mainly focus on Data Reconstruction Attacks (DRAs) that are considered the most severe privacy leakage. Existing attack techniques for reconstructing the original training data mainly fall into GAN-based class representation attacks and gradient-based instance reconstruction Attacks. Hitaj et al. [9] and Wang et al. [10] successively proposed DMU-GAN and mGAN-AI attack methods, in which an attacker can train a Generative Adversarial Network (GAN) against a target training set to generate samples. However, these methods require the attackers to have access to auxiliary raw data and require less diversity for all class members. In addition, the reconstructed samples are only class representations of the training samples, not the exact real data.

Recent efforts have turned on gradient-based data reconstruction attacks [11] [12], which relax the assumption on auxiliary data and allow pixel-level restoration, i.e., instance reconstruction. The main idea behind these methods is to optimize the "dummy" data (initialized with white noise) into real data via continuously minimizing the distance between the "dummy" gradient and the real gradient. However, these end-to-end approaches start with white noise for optimization, which suffers from poor convergence. This may be because the optimization process is unstable when the gradient values fluctuate without certain constraints.

Geiping et al. [13] and Yin et al. [14] alleviate this issue by adding some useful regularizations, e.g., batch normalization (BN) statistics (i.e., the mean and variance of batch training samples). The BN layer is usually used to normalize the batch samples during model training. With BN statistics, an attacker can apply the same normalization to one's recovery for better reconstruction. However, in a realistic FL setup, local clients usually do not share their pri-

vate BN statistics, dramatically reducing the scope of such reconstruction attacks. Experiments in [16] evaluated that relaxing the BN assumption can significantly weaken these attacks. GGL [15] combines gradient matching with a GAN trained on a public dataset. This approach is challenging to reconstruct images outside the prior distribution of public datasets. As a result, the reconstructions may suffer from information loss (e.g., change of image orientation or loss of key semantics).

Unlike previous end-to-end approaches, we propose a three-stage optimization framework named CGIR. In the label inference stage, we conduct a batch label restoration attack that allows multiple images to share the same labels. The inferred labels will be provided as conditional information to the subsequent image inference. Then, we conduct a "coarse-to-fine" image reconstruction process that can provide a stable and effective data reconstruction. Specifically, the coarse-level stage mainly focused on recovering the global layout of the image contents, like shapes, and structures of the contents through a conditional generator. The fine-level stage aims to refine the textual details of local areas in the reconstructed images by matching the gradients.

Different from the existing GAN-based approaches, the parameter update of our conditional generator does not depend on the discriminator trained from the auxiliary dataset, but on 1) the loss of matching the gradients and labels between the synthetic and original target images and 2) the Total Variance (TV) loss enhancing the stability of the optimization process. The encoding capacity of the generator provides a satisfyingly good holistic estimate of the image contents, which facilitates the fine-tuning of the following stage. Without the auxiliary data, CGIR can also avoid the limitation of the public data distribution on the reconstruction results. In addition, we equip the generator with inferred labels as prior information to ensure that the contents and the labels of the reconstructed images are consistent. Based on the recovered global structure, each pixel value of the synthetic image is updated directly in the fine-level stage, thus allowing a better and faster refinement of the image details. Previous methods that start directly with white noise usually require BN statistics to guide the optimization process. In our method, the global structure provides a satisfyingly good holistic estimate of the image contents, which helps to direct the optimization process.

We make the following contributions.

1) We introduce a novel instance reconstruction attack method based on a conditional generator, termed CGIR. It relaxes the auxiliary assumptions of auxiliary data and BN statistics, thus showing a more realistic and broader privacy leakage to FL than previous attacks.

2) We propose a batch label inference attack that allows multiple images to share the same labels in FL non-IID (identically and independently distributed) scenarios. Based on the inferred labels, we further introduce a "coarse-to-fine" image reconstruction process that can provide a stable and effective reconstruction. In addition, we add a conditional restriction to the restoration process to keep the constructed images always matching their corresponding true labels. The consistency of the reconstructed images' content and labels further exposes the original data's sensitive information.

3) We evaluate the effectiveness of our CGIR attack on two different network classifiers and five image datasets. Experimental results demonstrate that the CGIR attack is superior to prior arts, even for complex datasets, deep models, and large batch sizes. For example, on the CIFAR100 dataset, the CGIR attack can recover a considerable amount (about 40%) of the original visual information at a batch size of 168. We also evaluate several existing defenses against privacy breaches. Our experimental results suggest that pruning can be used to mitigate privacy risks in FL when a slight loss of model accuracy can be tolerated.

2 PRELIMINARIES

2.1 Federated Learning (FL)

An FL system coordinates a central server and multiple local clients to execute an iterative optimization procedure of model training. At each iteration, the server first randomly selects n clients and distributes the current joint model $f(\theta)$ to them. The selected clients then train the received model based on their local data and upload their model updates to the server. Finally, the server combines these model updates from the clients and changes the model for the next iteration. The iterative interactions between the server and the clients will continue until the model converge [17].

Consider an FL scenario with total clients $C_{i=1}^{N}$, where each client owns data $D_{i=1}^{N}$. The global learning objective is to minimize the weighted average of risks for all clients:

$$\min_{\theta} F(\theta) = \sum_{i=1}^{N} \frac{|D_i|}{|D|} L_i(\theta) \tag{1}$$

where $L_i(\theta)$ denotes the local empirical risk of each client C_i over their data D_i:

$$L_i(\theta) := E_{x_i \sim D_i}[l(f(x_i;\theta), y_i)] \tag{2}$$

$l(\cdot)$ is the loss function used by the local clients for training, such as cross-entropy. After training, the central server can obtain joint model parameters θ that can satisfy the F minimum without obtaining any private training data D_i.

There are various optimization algorithms proposed for aggregating local model updates based on client nodes. We review two mainstream FL architectures: Federated Stochastic Gradient Descent (FedSGD) and Federated Averaging (FedAvg) based on [18] and [19], respectively. At the tth iteration, the central server sends the current joint model M^t with the model parameter θ^t to each of the n chosen clients. For FedSGD, each client calculates and uploads the gradient updates g_i^t based on their local data. These gradient updates are aggregated by the server and are used to change the global parameter θ^{t+1} by the stochastic gradient descent (SGD) algorithm with learning rate η:

$$\theta^{t+1} = \theta^t - \eta g^t, g^t = \textstyle\sum_{i=1}^{n} \frac{|D_i|}{|D|} g_i^t \tag{3}$$

For FedAvg, each chosen client updates θ^t on their local data to obtain θ_i^{t+1}, and sends the updated parameter back to the server. The global parameter θ^{t+1} in next iteration will be calculated as:

$$\theta^{t+1} = \textstyle\sum_{i=1}^{n} \frac{|D_i|}{|D|} \theta_i^{t+1} \tag{4}$$

Note that even though the joint model is computed by the updated parameters from the clients, it is still possible for an attacker to derive the gradients by using successive snapshots of the joint model parameters [7].

2.2 Privacy Leakage in FL

Despite that FL avoids clients from disclosing their private data directly, recent studies have revealed that sensitive information about clients' private data is still at risk of leakage when sharing model updates. Several studies have attempted to explain why sharing gradients can cause privacy leakage of data [4] [20] [21]. A deep learning model can be viewed as a high-dimensional representation of the dataset it was trained on. The gradients of a specific layer are calculated based on the layer's features and the loss from the layer that follows it [22]. Any effective model may have recognized and memorized more data attributes than are necessary for the main learning objective, which makes the privacy leakage possible [23] [24] [25] [26] [27]. Based on the disclosure information, gradient-based privacy inference attacks can be broadly categorized as membership inference attacks, property inference attacks, and data reconstruction attacks.

The goal of membership inference attacks is to figure out whether a specific sample belongs to the training data [28] [29] [30] [31] [32] [33]. For example, in a deep natural language processing model trained on text input, non-zero gradients in the embedding layer can disclose which words have been used in the clients' training batches [7]. Property inference attacks aim to infer the sensitive privacy attributes from the training data, e.g., is the race of the training data black.

Data Reconstruction Attack. The data reconstruction attack aims to accurately reconstruct the original training samples. Hitaj et al. [9] proposed the first GAN-based data inference attack, called DMU-GAN. By exploiting the real-time interactivity of FL, DMU-GAN enabled the attacker to train a GAN against a specific category of the target training set to generate samples that he does not possess. However, it assumed that the attacker had auxiliary labels for the target data and required less diversity in the training data. A follow-up work [10] extended this approach to user-level privacy leakage, but still failed to get rid of the auxiliary data and merely generated a representation of the original data.

Recent efforts have focused on pixel-level detailed recovery without assuming auxiliary data. The main idea is to optimize the "dummy" data (initialized with white noise) into real data by minimizing the difference between the "dummy" gradient and the real gradient, i.e., gradient matching. Deep Leakage from Gradients (DLG) by Zhu et al. [11] presented a joint optimization formulation on the labels and input data via gradient matching. iDLG [12] facilitated the extraction fidelity by simplifying the objective function of DLG with the ground truth label computed analytically from the last layer of shared gradients. However, both DLG and iDLG used random noise as the initial point of optimization and suffered from poor convergence. This may be because the optimization process is unstable when the gradient values fluctuate without certain constraints. Moreover, these methods have been limited to shallow models and a small batch size of low-resolution image setups of less practical relevance (for a maximum batch size of 8 for DLG and a single label extraction for iDLG).

Inverting Gradients (IG) [13] adjusted the DLG's objective with cosine similarity and added total variation (TV) as a prior regularization, attaining some success in deep models and high-resolution images. GradInversion (GI) [14] introduced batch label restoration and made a breakthrough in larger batch reconstruction by regularizing image fidelity with group consistency. Although these approaches have had some success, they rely on a strong assumption that the attacker knows BN statistics. Relaxing this assumption can substantially reduce the effectiveness of these attacks, as demonstrated in [16]. GGL [15] combines gradient matching with a GAN trained on public datasets. Although the GAN model can help recover images, the reconstructed images are limited by the prior distribution, and thus challenging to reconstruct image samples outside the distribution.

Unlike previous end-to-end approaches, we propose a novel three-stage instance reconstruction attack based on a conditional generator, termed CGIR. Specifically, we first employ a generator to capture the global layout of the target images, called coarse-level inference. The encoding capacity of the generator provides a satisfyingly good holistic estimate of the image contents, which facilitate the fine-tuning of the following stage. In the fine-level stage, each pixel value of the synthetic image is updated directly, allowing a better and faster refinement of the image details. Note that CGIR drops the previous auxiliary assumptions, i.e., BN statistics and auxiliary data, thus showing a more realistic and broader privacy leakage to FL than previous attacks.

2.3 Privacy Defenses in FL

Two fundamental strategies to prevent privacy leakage of sensitive data in FL are encryption gradients and Perturbing gradients.

Encrypt Gradients. Existing works on encryption for gradients are typically based on previous cryptographic techniques, including Homomorphic Encryption (HE) [34] and Secure Multiparty Computation (SMC) [36]. However, the cryptography operations are not only time-consuming and resource-intensive, but also degrade the model's accuracy [35]. The implementation of SMC involves synchronized coordination among workers during training, which requires a high degree of stability in each client's equipment [37] [38] [39].

Perturbing Gradients. Another effective way to reduce private information leakage is to perturb as much of the valid information contained in the gradient as possible without affecting the model performance. One straightforward perturbation strategy is gradient compression, where gradients of small magnitude are pruned to zero, such that only a part of local updates will be communicated between devices and the server [11]. Abdelmoniem et al. [40] proposed a threshold-based compression scheme for distributed training systems, which does not affect the model performance even when the pruning ratio is 0.9. The other strategy is adding noise to gradients before sharing, thus confusing the information of the original data. McMahan et al. [41]

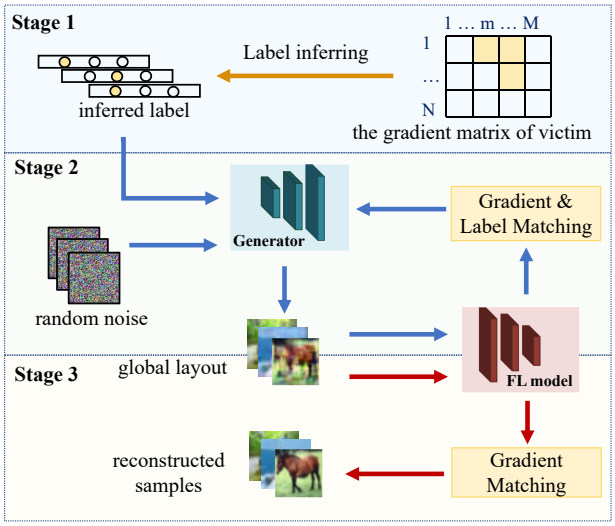


Fig. 1. Overview of our proposed CGIR attack. In stage 1, the labels of target images are inferred by analyzing the gradient sign of the last fully-connected layer. In stage 2, the generator decodes the global layout of the target images by matching both gradients and labels between reconstructed and true images. In stage 3, with the recovered global structure of the image contents provided by stage 2, a pixel-wise update is conducted through gradient matching.

proposed to add noise on the server side for LSTM language models. This ensures that malicious clients cannot infer or attack other benevolent clients. Another approach is for clients to add a degree of Gaussian or Laplacian noise before sharing their gradients, thus avoiding attacks from the malicious server side [11]. However, these techniques require a trade-off between defense capability and model performance, i.e., if the degree of perturbation to the gradient is too small, its defense capability will be poor, and conversely, the model performance will be compromised.

3 CGIR Attacks

In this section, we first describe the threat model of our CGIR. Then, we present the details of our attack pipeline, which consists of three stages: label inference, coarse-level inference, and fine-level inference.

3.1 Threat Model

Suppose that N local clients (where $N \geq 2$) jointly accomplish an FL task (i.e., image classification) under the coordination of a central server. The adversary of our CGIR attack can be an honest-but-curious server or a malicious eavesdropper in the communication channels between the clients and the server. We assume one of the clients is the victim client.

Adversary's goal: The adversary's primary goal is to recover the exact images that the victim possesses. It is a passive attack and does not affect the training process of the original model.

Adversary's knowledge: An adversary is allowed to store and process model updates transmitted by individual clients separately but will not interfere with the training algorithm. Unlike the previous work, we assume that the adversary does not access the original data's BN statistics during training to discuss a more realistic scenario.

Adversary's capabilities: Victims usually do not share the category labels of their uploaded gradients. However, an adversary can restore the ground truth labels by analyzing the gradients of the last fully-connected layer, as discussed in [12] [14]. Based on the prior arts, we further analyzed the label inference capability in the non-IID FL scenarios in Section 3.2.1.

3.2 Attack Pipeline

In this section, we describe our CGIR framework in detail, which can be roughly divided into three stages. In stage 1, we conduct a label inference attack to get the target images' corresponding labels by analyzing the sign of the gradients. The inferred labels will be provided as conditional information to the subsequent image inference.

In stage 2, we aim to recover the global layout of the image contents, like shapes or structures of the contents, through a conditional generator. In stage 3, we finely tune each pixel value of the synthetic images based on the holistic estimate of the target image contents. Figure 1 depicts the workflow of our CGIR attack, and the algorithm is summarized in Algorithm 1.

Algorithm 1 CGIR

Input: differentiable global model $f(x;\theta)$, global model parameters θ, gradients produced by local training data g, learning rate η, generator $G(w)$, the number of epochs T1 and T2 for stage 2 and stage 3, total variation function $\mathcal{TV}(\cdot)$, noise z sampling from $\mathcal{N}(0,1)$ as initial random vector inputs for Generator.

Output: reconstructed data $\hat{x}$.

Stage 1: Label Inference
  $y \leftarrow$ analyzing the last layer of g by Algorithm 2
Stage 2: Coarse-level Inference
  for i=0 to T1 **do**
    $\hat{g}_i \leftarrow \sum_l \nabla_{\theta^{(l)}} l(G(z,y;w_i),y)$;
    $\hat{y}_i \leftarrow f_\theta(G(z,y;w_i))$;
    $\mathcal{R}_{tv} \leftarrow \mathcal{TV}(G(z,y;w_i))$;
    $L_{sum} = \alpha_g||\hat{g}_i - g||_2 + \alpha_y||\hat{y}_i - y||_2 + \alpha_{tv}\mathcal{R}_{tv}$;
    $w_{i+1} \leftarrow \hat{w}_i - \eta L_{sum}$;
  $w = w_{i+1}$
  return $G(z,y;w)$ as $\hat{x}$;
Stage 3: Fine-level Inference
  for i=0 to T2 **do**
    $L'_{sum} = ||\nabla_\theta l(f(\hat{x}_i;\theta),y) - g||_2$;
    $\hat{x}_{i+1} \leftarrow \hat{x}_i - \eta L'_{sum}$;
  return $\hat{x}_{i+1}$;

3.2.1 *Stage1: Label Inference*

By analyzing the numerical distribution of the last layer of gradients, previous studies have demonstrated the feasibility of recovering labels. Considering a classification model with C categories, the last layer of the model is usually a fully connected layer (FC), which can be expressed as $b = \theta_{FC} r$. Where r is the input to the FC layer, θ_{FC} is the weight matrix, and b is the output. Given a batch size B

of images $x = [x_1, x_2, \cdots, x_B]$, the gradient of the loss $l(\cdot)$ with respect to θ_{FC} is:

$$g_{\theta_{FC}} = \frac{1}{B}\sum_i \frac{\partial l^i}{\partial b^i}\frac{\partial b^i}{\partial \theta_{FC}} = \frac{1}{B}\sum_i \frac{\partial l^i}{\partial b^i}(r^i)^T \quad (5)$$

For each image x_i, the $\frac{\partial l^i}{\partial b^i} = p_{i,c} - y_{i,c}$ at index c, where $p_{i,c}$ is the post-softmax value of model output in range $(0, 1)$, and $y_{i,c}$ is the value (0 or 1) of y_i at index c. Since the previous layer of the FC layer usually contains a common activation function (such as ReLU or Sigmoid), $(r^i)^T$ is always non-negative. Therefore, when $B = 1$, the values in θ_{FC} are negative only in the row where the ground truth label is located, while the values of other rows are positive. Zhao et al. [12] revealed this relationship, and presented an analytical method to extract the ground-truth label from the shared gradients with 100% accuracy. When the data distribution of each client in an FL scenario is extreme non-IID, i.e., there is only one category for each client, we observe that the gradients w.r.t. the last-layer weights also follow this rule.

For multi-sample batch training, the value of θ_{FC} is a linear summation from all images in this batch. The relationship between the ground truth labels and the negative sign of shared gradients may be diluted when the summation brings positive values from other images. Yin et al. [14] observed a more robust negative sign of the shared gradients for multiple image training. They only utilized the mth column where the minimum value of θ_{FC} is located, instead of all rows of θ_{FC}. The indexes of the top-B minimum values in column m are the inference labels.

The top-B minimum in the mth column of the gradient matrix θ_{FC} may include positive values when there are repeating labels in a batch. That is contrary to the relationship between labels and signs of gradients. Based on this observation, we refine the existing approaches and design a batch label restoration method that multiple images can share the same labels. Specifically, we first locate the column m of θ_{FC} and only record the indexes of rows with negative values as y. When the number of negative values is greater than B, we take top-B indexes as the inferred labels. Otherwise, we remove the column m of $g_{\theta_{FC}}$ and make the updated $g_{\theta_{FC}}$ as the new $g_{\theta_{FC}}$. Then, We repeat the first step until the length of the inferred y is equal to B. The algorithm is summarized in Algorithm 2.

3.2.2 Stage2: Coarse-level Inference

After recovering the image's labels, we utilize a conditional generator to capture the global structures of the target images. The generator can be seen as a neural network with feature extraction capability, which takes random noise and inferred labels as model input and outputs the synthetic images. The objective function of optimizing the generator's parameters includes 1) the loss of matching the gradients and labels between the synthetic and original target images and 2) the Total Variance (TV) loss enhancing the stability of the optimization process. Minimizing the gradient and label differences between the synthetic images and original target images aims to enforce the synthetic images close to the ground truth. When the optimization finishes, the global layout of the image contents, like the shapes and structures of the target images, will be covered.

Algorithm 2 Label Inference

Input: g: the gradients produced by local training data; B: the number of batch size.
Output: the inferred labels of target images y.

```
get the g_{θ_FC} from g
b = 0, y = []
while b < B do
    g^m_{θ_FC} ← locate the mth column where the minimum value of the g_{θ_FC} is located.
    y_i ← sort the values of g^m_{θ_FC} in ascending order and record the indexes of rows with negative values.
    Append y_i to y.
    b = b + len(y)
    if b > B then
        y = y[: B]
    g_{θ_FC} ← update the g_{θ_FC} after removing the mth column.
```

Given a batch randomly initialized noise z ($z \in R^{B\times C\times H\times W}$, B, H, W, C being the batch size, height, width, and image channels) and the inferred labels y. The optimization goal of stage 2 is illustrated as follows:

$$\min_{\hat{w}} \mathcal{L}_g(\hat{g}, g) + \mathcal{L}_y(\hat{y}, y) + \mathcal{R}_{tv}(G(w; z, y)) \quad (6)$$

Where g is and gradients of target images; $G(w; z, y)$ are the synthesized images decoded by generator, hereinafter called $\hat{x}$; $\hat{y}$ and $\hat{g}$ are extracted labels and corresponding gradients of the synthesized images $\hat{x}$; $\mathcal{L}_g(\cdot)$ and $\mathcal{L}_y(\cdot)$ perform the gradient and label matching for the synthetic and real data; $\mathcal{R}_{tv}(\cdot)$ is an image prior regularization that provides a more stable convergence to this process [42].

For gradient matching, we minimize the ℓ_2 distances between gradients on the ground truth images x and synthesized images $\hat{x}$:

$$\mathcal{L}_g(\hat{g}, g) = \alpha_g \sum_k ||\nabla_{\theta^{(k)}} l(f(\hat{x}; \theta), y) - g^{(k)}||_2 \quad (7)$$

Where $g^{(k)}$ and $\nabla_{\theta^{(k)}} l(f(\hat{x}; \theta), y)$ refer to gradients on the real images and synthesized images at layer k, respectively. All layers are summed and scaled by α_g. For label matching, we penalize the ℓ_2 norm of labels on the real images and the synthesized images $\hat{x}$ and scale it with α_y:

$$\mathcal{L}_y(\hat{y}, y) = \alpha_y ||f_\theta(\hat{x}) - y||_2 \quad (8)$$

This label restriction ensures that the content and categories of the reconstructed images are consistent.

During image restoration, noise may cause gradient explosion and thus affect the stability of the optimization process, especially when the gradient value fluctuates drastically. The total variation (TV) loss encourages spatial continuity and smoothness in the synthetic images by reducing the difference between adjacent pixel values. The definition of TV loss is given as follows:

$$\mathcal{R}_{tv}(\hat{x}) = \alpha_{tv} \sum_{i,j} ((\hat{x}_{i,j+1} - \hat{x}_{i,j})^2 + (\hat{x}_{i+1,j} - \hat{x}_{i,j})^2)^{\frac{\beta}{2}} \quad (9)$$

Where β can be used to adjust the image continuity. Setting the β to 2 usually gives a good trade-off between image smoothing and preservation of image details, as demonstrated in [42]. We follow the prior work setting the $\beta = 2$, the reconstructed images are smoothed enforced by this loss.

Two aspects need to be emphasized: 1) The target of the objective function is not the image's pixels but the generator's parameters. A well-trained generator can provide a smooth latent space for image decoding. 2) Compared to the standard generator, the generator in our CGIR incorporates inferred labels as the condition information, which are also added to the objective function as label regularization. This label regularization term allows the generator to differentiate between different image categories, thus ensuring the contents and the classes of the reconstructed images are consistent.

3.2.3 Stage3: Fine-level Inference

With the recovered global structure of the image contents provided by stage 2, further fine details are completed in the third stage. In this stage, we conduct a pixel-wise update by matching the gradients. The objective function can be defined as follows:

$$\min_{\hat{x}} \sum_k ||\nabla_{\theta^{(k)}} l(f(\hat{x};\theta), y) - g^{(k)}||_2 \tag{10}$$

Stage 3 starts from the image's global layout obtained in stage 2, rather than from noise. The global structure provides a satisfyingly holistic estimate of the image contents and therefore facilitates the fine-tuning at this stage. In addition, each pixel value of the synthetic image is updated directly, allowing a better and faster refinement of the image details.

Note that the fine-level stage skips the auxiliary generator so that the computational cost is lower than the coarse-level stage. But if only the fine-level stage is used, it usually causes instability in the model optimization process. Therefore, by balancing the scale of these two steps, CGIR attacks can further reduce computational costs while maintaining good image quality. We will discuss the trade-off between these two stages in our ablation study.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our attack on multiple datasets: 1) the grayscale handwritten digit images of 28×28px (MNIST) [43], 2) the CIFAR10 dataset [44] and CIFAR100 dataset [45] of 32×32px RGB images, 3) the CalabFaces Attributes detaset of 64×64px (CelebA-HQ) [46], and 4) the Imagenet dataset of 128×128px [47]. More details of these selected datasets are listed in Table 1. For CelebA-HQ dataset, we segmented the dataset using male and female as category attributes, following [48]. The selected datasets differ in sample resolution, size of the dataset (total number of samples), and number of categories. This allows us to examine the threats posed by attacks in different dataset complexity, with the first three low-resolution datasets being basic and the remaining two high-resolution datasets being complex.

Models. We adopt two networks as image classifiers to discuss privacy risks at varying model complexity. Following the first gradient-based reconstruction attack (DLG), we use the same shallow and smooth LeNetZhu [11] to explore the availability of our attack methods. To ensure the twice-differentiable of the model, LeNetZhu replaced the activation ReLU with Sigmoid and removed the strides. We then focus on the deep ResNet-18 model as the backbone network to explore the performance of our attack in complex architecture. Since the L-BFGS algorithm requires second-order differentiability, we use the ELU activation function for all attack methods, which can reduce the gradient vanishing while retaining the ability of non-linearity.

TABLE 1
Dataset characteristics.

Dataset	Resolution	Size	Category
MNIST	28×28×1	60000	10
CIFAR10	32×32×3	60000	10
CIFAR100	32×32×3	60000	10
CelebA-HQ	64×64×3	23705	2
ImageNet	128×128×3	1281167	1000

The generator of our CGIR is fed with random noise z and the inferred labels y that are extracted by analyzing the last layer of shared gradients. The noise z for the generator is derived from a 128-dimensional standard normal distribution with mean 0 and variance 1. The generator consists of two input embedding layers, followed by a view function to reshape the feature map, where MNIST is set to 7×7, and the other datasets are set to 4×4. It then goes through several upsampling blocks to improve the spatial resolution of feature maps, as in [48]. Depending on the dataset, the number of upsampling blocks varies. To increase the smoothness of the generated images, we use a Sigmoid activation layer before the model output. In upsampling block, we test both nearest-neighbor interpolation and ConvTranspose2d to resample the feature map. Both upsampling methods work well with our framework in our experiments, and due to space limitations, we present the experimental results with the first structure. More details about the training models are provided in Appendix A.

Implementation Details. An FL system is usually set up with multiple participants with non-IID data distributions. We presume there are 100 clients in total, and 10 of them are selected randomly in each round. Among the selected clients, one of them is the victim. To simulate non-IID data settings, we use a Dirichlet distribution with the haperparameter γ to divide data for different clients. In general, the smaller the value of haperparameter γ, the higher the degree of non-IID distribution of the data. Following [49], we set the $\gamma = 0.9$ for our comparison studies.

Since the goal of the CGIR attack is instance reconstruction, it is not fair to compare the GAN-based class representation attack with it. Therefore, our current comparison focuses on iDLG and IG. iDLG applied the ℓ_2 cost function with the L-BFGS optimizer, and IG relied on a combination cost function of cosine similarity and TV regularization with the Adam optimizer. The generator in our CGIR framework is trained by RMSprop optimizer with a learning rate of 1e-2 and momentum of 0.9. We take the best hyper-parameter of α_g = 1, α_y = 1e-2 and α_{tv} = 1e-6.

Evaluation Metrics We evaluate the quality of the reconstructed images from both qualitative and quantitative perspectives. The visual fidelity of the reconstructed images to the real images can be used as an indicator of perceptible image similarity. The quantitative metrics include: 1) the

Mean Square Error (MSE ↓), 2) the Peak Signal-to-Noise Ratio (PSNR ↑), and 3) the Structural Similarity Index Metric (SSIM ↑) between real and reconstruction images [50]. (The ↑ and ↓ correspond to the higher or lower values of the corresponding metrics when the constructed image is closer to the real image, respectively.)

MSE represents the mean squared euclidean distance between the actual and reconstructed images. Given two images x and y of size $m \times n$, the function of MSE is defined as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [x(i,j) - y(i,j)]^2 \tag{11}$$

A lower MSE score indicates a higher similarity between the two images.

PSNR is an objective criterion for evaluating the similarity between a reconstructed image and a real image, which is defined as the logarithm of the ratio of the maximum squared value of image fluctuations to the MSE between two images. The formal definition is given in Equation (12):

$$PSNR = 10 \cdot log_{10}(\frac{MAX_I^2}{MSE}) \tag{12}$$

where MAX_I^2 denotes the maximum possible pixel value of the image. In general, the higher the PSNR value, the smaller the distortion between the estimated and the real image, and the better the image quality.

SSIM is to measure the similarities between two images from the perspective of their composition. Given two images x and y, the structural similarity of the two images can be calculated as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{13}$$

Where μ_x and μ_y, σ_x and σ_y, and σ_{xy} denotes the mean values, variance and covariance of x and y, respectively. $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ are constants used to maintain stability and L is the dynamic range of the pixel values, where $k_1 = 0.01, k_2 = 0.03$. The range of structural similarity is from 0 to 1. When two images are identical, the value of SSIM is equal to 1.

4.2 Comparison Studies

In this section, we analyze the attack capability of our CGIR without BN statistics assumption and compare it with iDLG and IG under different task complexities. We first use a shallow smoothing model (LeNetZhu) to test the feasibility of our CGIR in the single image reconstructions, similar to previous studies on privacy breaches in FL settings. In the following, we focus on a realistic setup: a deep ResNet-18 classifier trained with a batch of RGB images. Finally, we show the performance of our CGIR attack in recovering a large batch of images from their averaged gradients. For all the following experiments, we obtain the labels of the target images by the label inference method in Section 3.2.1.

Results on LeNetZhu. We compare the performance of different attacks on LeNetZhu using five different datasets mentioned above. Since iDLG only supports the reconstruction of a single image, we set the batch size to 1. The quantitative comparisons are summarized in Table 2, and the best-performing visualization samples are provided in Figure 2.

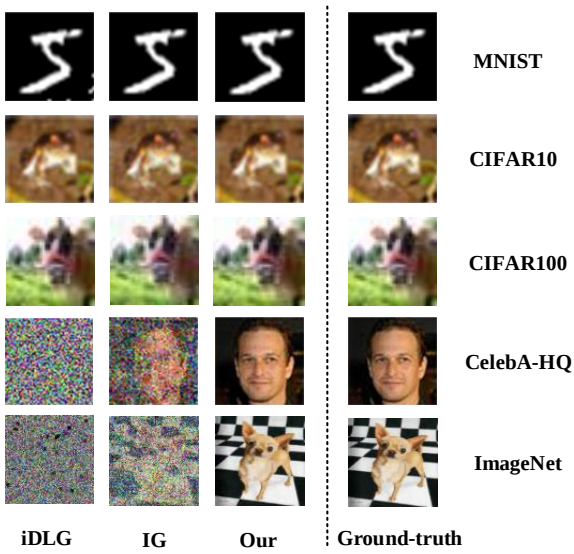


Fig. 2. Qualitative comparison on LeNetZhu.

As shown in Table 2, our CGIR attack is on par with the SOTA methods under the low-resolution datasets. For the MNIST dataset, the mean value of PSNR, SSIM, and MSE metrics for all methods are around 50, 0.99, and less than 0.001, respectively. Our attack method outperforms prior arts by a large margin for complex datasets. The mean values of PSNR and SSIM of CGIR attacks maintain at 32.71 and 0.87, while benchmarks are below 15 and 0.2, respectively. This is because that previous end-to-end benchmarks directly optimize the pixel values of white noise to natural images, which may have difficulty in convergence, especially when the distribution of the images is complex, such as the extremely diverse nature of the image content. Our CGIR combines the stability of global structural optimization with the accuracy of local detail fine-tuning and thus can perform well even on complex datasets.

From the results presented in Figure 2, we observe that all the attack methods produce identifiable images on the low-resolution dataset. Our method shows more remarkable performance on the CelebA-HQ and Imagenet datasets than the prior arts. Specifically, iDLG does not recover the images at all, and IG shows only some similarity to ground truth images in color distribution and geometry. However, our method still produces high-fidelity images with rich details. These visual results are consistent with the numerical results in Table 2. Therefore, we confirm that the CGIR attacks outperform the priors under the shallow models, even without auxiliary assumptions.

Results on ResNet. We now turn to a more realistic setting: a deep ResNet18 classifier and a batch of RGB images. Since iDLG is limited to shallow networks, we focus on the comparison experiments with IG. Table 3 presents the numerical performance comparisons between CGIR and IG attacks at a batch size of 8.

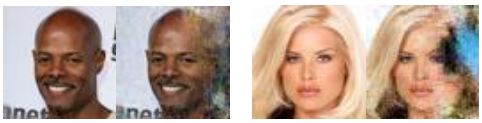
For the basic datasets, our CGIR attack shows a certain advantage regarding the std of image quality statistics, where the std of PSNR is only 0.82 for CIFAR10 and 1.76

TABLE 2
Quantitative comparison on LeNetZhu at batch size 1.

	MSE ↓			PSNR ↑			SSIM ↑		
	iDLG	IG	CGIR	iDLG	IG	CGIR	iDLG	IG	CGIR
MNIST	1e-3±6e-3	2e-4±4e-4	1e-4±5e-6	49.40±2.10	50.34±1.41	50.52±1.21	0.99±2e-3	0.99±5e-3	0.99±6e-6
CIFAR10	7e-6±2e-6	4e-3±5e-3	5e-4±6e-4	51.52±1.26	28.45±6.93	34.60±4.03	0.99±1.07	0.84±0.15	0.97±0.03
CIFAR100	0.07±0.11	0.01±8e-2	4e-3±2e-4	24.98±8.10	27.8±5.33	29.54±3.23	0.59±0.46	0.76±0.13	0.81±0.01
CelebA	0.05±1e-3	0.05±0.01	**0.01±0.01**	12.39±0.08	12.45±1.10	**28.94±4.12**	0.01±0.01	0.15±0.03	**0.84±0.31**
Imagenet	0.15±0.07	0.08±4e-3	**5e-3±1e-3**	8.46±1.59	11.61±0.23	**32.71±2.81**	0.03±0.03	0.01±6e-3	**0.87±0.21**

TABLE 3
Quantitative comparison on ResNet18 at batch size 8.

Dataset	Metric	IG		CGIR	
		Mean	Std	Mean	Std
CIFAR10	MSE ↓	0.06	0.01	**1e-5**	**4e-6**
	PSNR↑	33.36	11.48	**44.11**	**0.82**
	SSIM↑	0.78	0.21	**0.99**	**2e-5**
CIFAR100	MSE ↓	0.01	0.01	**3e-3**	**5e-4**
	PSNR↑	32.13	6.31	**33.92**	**1.76**
	SSIM↑	0.66	0.23	**0.99**	**1e-4**
CelebA-HQ	MSE ↓	0.07	0.02	0.06	0.02
	PSNR↑	11.33	1.64	15.51	2.56
	SSIM↑	0.15	0.06	0.38	0.22
ImageNet	MSE ↓	0.05	0.02	0.06	0.02
	PSNR↑	12.82	1.93	12.91	2.13
	SSIM↑	0.14	0.07	0.23	0.08



(a) details leakage

(b) gender leakage

Fig. 3. Face details (a) and gender information (b) leakage at batch size 8 on CelebA-HQ. Each pair of comparison images contains an original sample (left), and a reconstructed image by CGRA (right).

for CIFAR100, while the std of PSNR for IG on CIFAR10 and CIFAR10 are 11.48 and 6.31, respectively.

For CelebA-HQ and ImageNet datasets, the performance of both CGIR and IG is impaired, but CGIR is still slightly better than IG. Figure 3 and Figure 4 present the varying degrees of information leakage of our method on the CelebA-HQ and Imagenet datasets, respectively. In the case of CelebA-HQ, it is possible to identify a person's gender or specific facial features, despite partial location blurring. For the more complex ImageNet, the background information of the images is leaked. These results highlight the risk of data leakage caused by our attack under complex models and datasets.

Large-batch Images Recovery. We now increase the batch size to compare the upper limit of the number of images recovered by IG and CGIR. For both CIFAR10 and CIFAR100 datasets, we test the batch size of 16, 32, 64,

Fig. 4. Background leakage at batch size 8 on Imagenet. The first row shows the original images and the second row shows the reconstructed images by CGRA.

100, 128, and 168. We report the best results for PSNR, SSIM, and MSE between a single reconstructed image and the corresponding real image in different batch sizes, as in Figure 5. More comparisons of reconstruction results are provided in Appendix B.

We observe that IG faces difficulties in recovering images when the batch size is 32, as PSNR values are below 25 and SSIM values are below 0.7. However, CGIR, at batch sizes of 64 and 168 for CIFAR10 and CIFAR100, respectively, have a PSNR of 30 and an SSIM of 0.9. The visualized attack results show that the number of recognizable images drops as the number of images corresponding to the average gradients increases. Surprisingly, CGIR still restores a decent amount of original visual information at batch size 64 on CIFAR10 (about 20%) and batch size 168 on CIFAR100 (about 40%). The entire batch's reconstruction results are provided in Appendix B.

Figure 6 and Figure 7 show a case study of the multi-image recovery of our CGIR attack for CIFAR10 and CIFAR100 datasets, respectively. As expected, with the increase in batch size, the information leakage of a single image can be mitigated to some extent. For the CIFAR10 dataset, the MSE values of the reconstructed images increased from 4.9e-5 to 6.4e-3, the PSNR values decreased from 43.05 to 21.67, and the SSIM values decreased from 0.99 to 0.90 as batch size grew. The CIFAR100 dataset with larger batch sizes shows a similar reconstruction performance. Therefore, we can confirm that our CGIR attack still poses privacy risks in high-volume image restoration, even without relying on BN statistics.

4.3 Ablation Studies

In this section, we investigate the effectiveness of the label restriction in the generator and the coarse-to-fine framework of our CGIR attack. We use ResNet18 as the classifier backbone for all experiments.

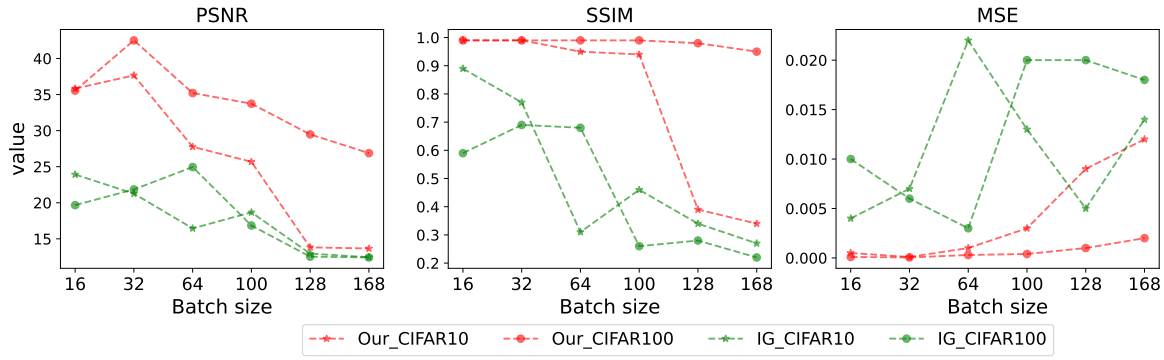
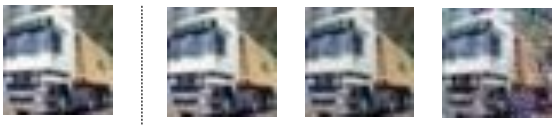
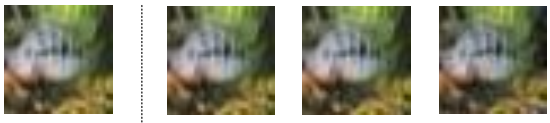


Fig. 5. Comparison results of different batch sizes on both CIFAR10 and CIFAR100.



MSE	1.7e-4	1.0e-3	3.0e-3
PSNR	37.66	27.76	25.68
SSIM	0.99	0.95	0.94

Fig. 6. Reconstruction images at different batch sizes on CIFAR10, where the leftmost image is the ground truth, followed by reconstructed images at batch size 32, 64 and 100 from left to right.



MSE	1.0e-4	1.0e-3	2.0e-3
PSNR	33.73	29.48	26.88
SSIM	0.99	0.98	0.95

Fig. 7. Reconstruction images at different batch sizes on CIFAR100, where the leftmost image is the ground truth, followed by reconstructed images at batch size 100, 128, and 168 from left to right.

4.3.1 Label Restriction

In the coarse-level stage of CGIR, we equip the generator with inferred labels as prior information, which is also added to the objective function as label matching loss. In this section, we test the reconstruction results of CGIR with and without label restrictions. Figure 8 shows a case study of the reconstructed images on the CIFAR100 dataset at batch size 8. As shown in Figure 8, if the optimization process contains label information, the content of the generated images is consistent with their labels, and conversely, the reconstructed images are disordered. This is because the uploaded gradients are averaged over the entire batch of images. A batch of N images has $N!$ different permutations with the same batch-averaged gradients. As a result, the reconstruction results of multiple images usually cannot correspond to the labels. However, if the optimization process is equipped with label restrictions, the generator can decode the images according to the label order, thus ensuring that the content of the reconstructed images and the labels are consistent. The consistency of the reconstructed images' content and labels further exposes the original data's sensitive information. Note that label matching loss is not required in the fine-level stage because the image's global layout obtained in the coarse-level stage have contained the label information.

4.3.2 Coarse-to-fine Balance

We now investigate the impact of the coarse-level stage and fine-level stage on the effectiveness of our CGIR attacks. We evaluate the performance of the attack using only the coarse-level stage, only the fine-level stage, and a combination of both in different proportions. Taking CIFAR100 as an example, we set the batch size to 8 and fix the total number of iterations to 300, which is much less than the number of iterations required by the first-order optimization method (e.g., IG).

Figure 9 shows the performance of our coarse-to-fine framework in different scale combinations. Where 'i_mse', 'i_psnr', and 'i_ssim' denote the difference between the reconstructed images and the original images in terms of different metrics. When only the coarse-level stage is included, the attack yields an acceptable result with a mean PSNR of 33.53 for the reconstructed images. However, the attack shows the worst performance when only the fine-level stage is included, where the average PSNR value of the images is only 11.91. Surprisingly, by combining these two stages, the PSNR value of the reconstructed image can reach up to 38.54, which exceeds the performance of the attack using only the coarse-level stage.

This is because using random noise as the initial point for optimization may lead to gradient explosion or falling into other local extremes, rendering the attack unstable and making it challenging to reconstruct the data points with high precision. Applying a generator first provides a satisfyingly good holistic estimate of the image contents, which facilitates fine-tuning in the following stage. In the fine-level stage, each pixel value of the synthetic image is updated directly, allowing a better and faster refinement of the image details. Therefore, combining the coarse-level and fine-level stages allows for a stable and accurate attack. Note that the fine-level stage skips the auxiliary generator so that the computational cost is lower than the coarse-level stage. Therefore, by balancing the scale of these two steps, CGIR attacks can further reduce computational costs while maintaining good image quality.

Figure 10 shows a case study of our CGIR attack with different scales of two stages for CIFAR100 dataset at batch

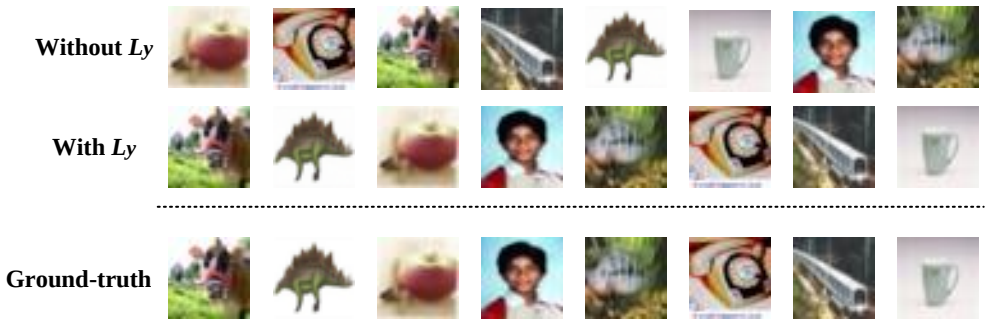


Fig. 8. Reconstruction images with and without label condition restrictions on the CIFAR100 dataset at batch size 8.

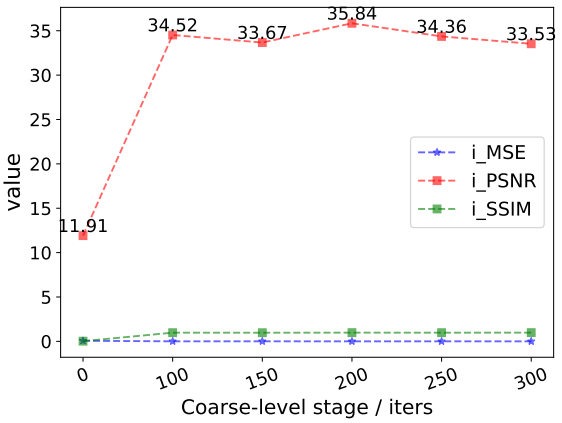


Fig. 9. Reconstruction under different proportions of the coarse-to-fine framework with a total number of 300 iterations.

size 8. As we can see, the reconstructed images do not contain any useful information but noise, if only the fine-level stage is conducted. When only the coarse-level stage is provided, the reconstructed images are smooth overall, but some texture information is blurred. When these two stages are combined, the reconstructed images have more texture information (e.g., the brighter feathers of the peacock in the 2nd image, or the clearer outline of the boat in the 4th image), even though there may be artifact pixels.

5 Attacks under Defence Strategies

Since encryption-based protection schemes always incur extra sophisticated setups and are costly to implement, we mainly evaluate the defense strategy of perturbing gradients in our experiments. The main purpose of this section is to measure the trade-off between the model accuracy and defendability under existing defense strategies against CGIR attacks.

Following prior study [11], we evaluate our CGIR attacks by pruning gradients and adding noise to gradients with the same setup. We first test Gaussian and Laplacian noise (extensively used in differential privacy researches) distributions with standard deviation n of 1e-4, 1e-3, 1e-2, 5e-2 and 1e-1 and center 0. Since the defense capability is less dependent on the type of noise [11], due to the limitation of space, only the defense results under Gaussian noise are shown in this paper. Then, we prune the gradients at ratios p of 0.1 to 0.9. We set the batch size set to 8, 16 and 32, and evaluate our method on CIFAR100 dataset using ResNet-18 backbone.

TABLE 4
The trade-off between model accuracy and defendability under different defenses at a batch size of 32.

Noise scale	Acc/DAcc	Defense	Pruning ratio	Acc/DAcc	Defense
1e-4	0.64/0.0027	No	0.2	0.64/0.0011	No
1e-3	0.64/0.0056	No	0.6	0.64/0.0024	No
1e-2	0.53/0.11	No	0.7	0.64/0.0035	No
5e-2	0.14/0.50	Yes	0.8	0.61/0.035	No
1e-1	0.04/0.60	Yes	0.9	0.55/0.095	Yes

Figure 11 shows the reconstructed images of the CGIR attack under two defenses, where for the pruning defense strategy, only results with p of 0.2, 0.6, 0.7, 0.8, and 0.9 are shown. We select the same images in different batches for visualization as a comparison. Table 4 shows the trade-off between model accuracy and defensibility for the two defenses with a batch size of 32, where 'Acc' represents the model accuracy under the defenses, 'DAcc' represents the corresponding decrease in model accuracy, and 'Yes' represents it successfully defends against CGIR attack while 'No' means failure to defend.

For the strategy of adding noise to gradients, our reconstructed images are still recognizable even at the standard deviation of noise of 1e-2 (see Table 11). The noisy gradients mitigate information leakage only when the variance is greater than 5e-2. However, in this case, the performance of the global model is severely affected, and the accuracy of the model drops to 0.14 (see Table 4). It suggests that adding noise to the gradient is insufficient to prevent data leaking in practice.

For the strategy of pruning gradients, the prune ratio suggested in [11] (20%) fails completely for CGIR attacks. As shown in Figure 11, the reconstructed images become blurred and dark as the pruning ratio increases. However, the reconstructions are still identifiable even at a pruning ratio of 0.9 with batch sizes of 8 and 16. The reconstructed images are almost difficult to identify until at a pruning rate of 0.9 with a batch size of 32. As a trade-off, the model's accuracy is reduced by about 10% at this point. (see Table 4). It suggests that pruning can be used to mitigate privacy risks in FL when a slight loss of model accuracy can be

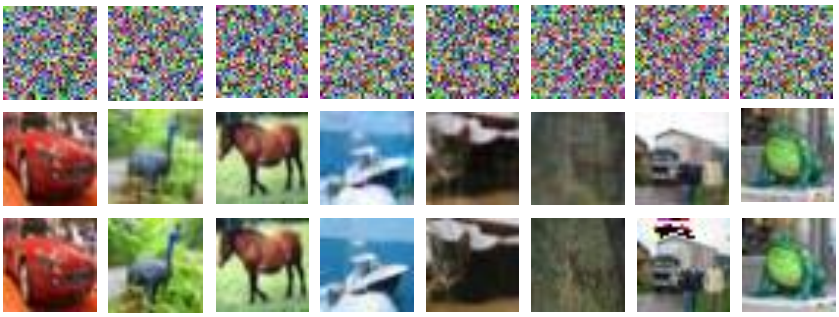


Fig. 10. A case study of CGIR attacks for coarse-to-fine balance. The first row shows the reconstruction results for the fine-level stage only. The second row shows the reconstruction results for the coarse-level stage only. The third row shows the reconstruction results when 200 iterations are executed in the coarse-level stage, and 100 iterations are executed in the fine-level stage.

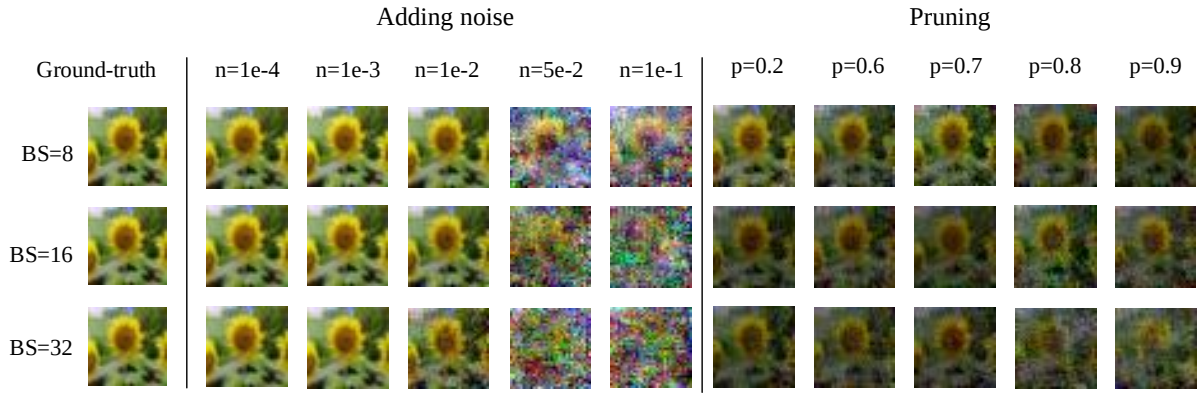


Fig. 11. Reconstruction images under different defenses with batch sizes of 8, 16 and 32.

tolerated.

6 Discussion and Analysis

In this section, we discuss the impact of different numbers of clients and non-IID data distributions in the FL settings on CGIR.

Number of clients. The key factor in launching a CGIR attack is to obtain the gradients uploaded by the victim client in a training round. An adversary can steal the victim's gradients, whether he is an honest but curious server or a malicious eavesdropper, and this procedure is independent of the number of clients. When the number of clients is 2, the adversary can also be one of the clients while the other is the victim. In this case, the adversary can save the snapshots of the joint model parameters. The difference between the consecutive snapshots is equal to the aggregated gradients from all participants. The adversary thus can subtract his own gradients from the aggregated gradients to get the victim's gradients. Without loss of generality, we focus on the case where the server or the eavesdropper is the adversary, and the number of clients does not affect the experimental results.

non-IID data distributions. The different non-IID data distributions is an important setting in the FL scenarios. We now investigate whether CGIR can still achieve good reconstruction performance when the Dirichlet haperparameter is raised to 1.5. We use ResNet-18 as the classifier backbone with both CIFAR10 and CIAFR100 datasets. Table 5 depicts the results of CGIR with batch sizes of 8 and 16. As we can see, CGIR behaves similarly at different Dirichlet parameters of 0.9 and 1.5, where the difference between the values of PSNR is around 1 and the values of MSE and SSIM are almost the same. Therefore, CGIR exhibits practicability and robustness when attacking FL.

TABLE 5
Quantitative comparison in different non-IID settings with Dirichlet haperparameter $\gamma = 0.9$ and $\gamma = 1.5$.

Dataset	Batch size	MSE↓		PSNR↑		SSIM↑	
		0.9	1.5	0.9	1.5	0.9	1.5
CIFAR10	8	1e-5	1e-5	44.11	45.34	0.99	0.99
	16	2e-4	5e-4	35.82	33.76	0.99	0.99
CIFAR100	8	2e-4	2e-4	33.92	34.02	0.99	0.99
	16	2e-4	2e-4	35.55	36.53	0.99	0.99

7 Conclusion

FL is a distributed learning paradigm that brings privacy benefits to users and drives the growth and deployment of artificial intelligence. However, there are still privacy risks of data leakage in the FL training process. Although the existing data reconstruction attacks have shown some effective performance, they make different assumptions about the settings. In this paper, we relax these assumptions and propose a conditional generative instance reconstruction attack, termed CGIR, which presents a more realistic and

broader privacy leakage to FL than previous attacks. Experimental results show that our CGIR attack is superior to prior arts, even for complicated datasets, deep models, and large batch sizes. In addition, the reconstructed images always match their corresponding real labels with label condition restriction, which further exposes the reconstructed data's sensitive information. We also evaluate several existing defenses and find that the effectiveness of current defense methods is based on the compromise of model accuracy.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China, under Grant U22B2027, U21A20463, and in part by the Fundamental Research Funds for the Central Universities of China under Grant KKJB320001536.

REFERENCES

[1] P. Liu, X. Xu, and W. Wang, "Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives," *Cybersecurity*, vol. 5, no. 1, pp. 1–19, 2022.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] H. Yang, J. Zhao, Z. Xiong, K.-Y. Lam, S. Sun, and L. Xiao, "Privacy-preserving federated learning for uav-enabled networks: Learning-based joint scheduling and resource management," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3144–3159, 2021.

[4] L. Lyu, H. Yu, X. Ma, L. Sun, J. Zhao, Q. Yang, and P. S. Yu, "Privacy and robustness in federated learning: Attacks and defenses," *arXiv preprint arXiv:2012.06337*, 2020.

[5] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Computing Surveys (CSUR)*, 2021.

[6] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *International conference on machine learning*. PMLR, 2021, pp. 1964–1974.

[7] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 691–706.

[8] M. Shen, H. Wang, B. Zhang, L. Zhu, K. Xu, Q. Li, and X. Du, "Exploiting unintended property leakage in blockchain-assisted federated learning for intelligent edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2265–2275, 2020.

[9] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 603–618.

[10] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2512–2520.

[11] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 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?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 937–16 947, 2020.

[14] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 337–16 346.

[15] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 132–10 142.

[16] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora, "Evaluating gradient inversion attacks and defenses in federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7232–7241, 2021.

[17] X. Lyu, Y. Han, W. Wang, J. Liu, B. Wang, J. Liu, and X. Zhang, "Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning," in *Proceedings of Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023)*, Feb 7-14, 2023, Washington DC.

[18] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[20] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning: Revisited and enhanced," in *International Conference on Applications and Techniques in Information Security*. Springer, 2017, pp. 100–110.

[21] A. Hannun, C. Guo, and L. van der Maaten, "Measuring data leakage in machine-learning models with fisher information," in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 760–770.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[23] L. Lyu, "Privacy-preserving machine learning and data aggregation for internet of things," *Ph. D. dissertation*, 2018.

[24] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 267–284.

[25] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[26] X. Liu, J. Liu, S. Zhu, W. Wang, and X. Zhang, "Privacy risk analysis and mitigation of analytics libraries in the android ecosystem," *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1184–1199, 2019.

[27] W. Wang, Y. Shang, Y. He, Y. Li, and J. Liu, "Botmark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors," *Information Sciences*, vol. 511, pp. 284–296, 2020.

[28] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[29] 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 *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[30] K. Leino and M. Fredrikson, "Stolen memories: Leveraging model memorization for calibrated {White-Box} membership inference," in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1605–1622.

[31] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv:1806.01246*, 2018.

[32] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 259–274.

[33] X. He, R. Wen, Y. Wu, M. Backes, Y. Shen, and Y. Zhang, "Node-level membership inference attacks against graph neural networks," *arXiv preprint arXiv:2102.05429*, 2021.

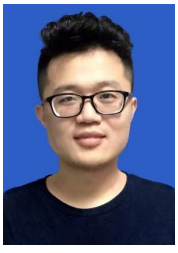
[34] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[35] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–6.

[36] A. C. Yao, "Protocols for secure computations," in *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE, 1982, pp. 160–164.

[37] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.
[38] W. Wang, J. Song, G. Xu, Y. Li, H. Wang, and C. Su, "Contractward: Automated vulnerability detection models for ethereum smart contracts," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1133–1144, 2020.
[39] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.
[40] A. M Abdelmoniem, A. Elzanaty, M.-S. Alouini, and M. Canini, "An efficient statistical-based gradient compression technique for distributed training systems," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 297–322, 2021.
[41] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.
[42] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
[43] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.
[44] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
[45] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia computer science*, vol. 132, pp. 377–384, 2018.
[46] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.
[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
[48] X. Xu, Y. Li, and C. Yuan, "Conditional image generation with one-vs-all classifier," *Neurocomputing*, vol. 434, pp. 261–267, 2021.
[49] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.
[50] D. R. I. M. Setiadi, "Psnr vs ssim: imperceptibility quality assessment for image steganography," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 8423–8444, 2021.

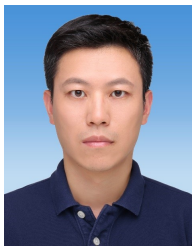
Xiangrui Xu received the BA and MA degrees in 2018 and 2021, respectively, at Wuhan Polytechnic University. She is currently pursuing a Ph.D. degree at Beijing Jiaotong University, China. Her research interests lie in trustworthy and interpretable AI technologies for cybersecurity applications.



Pengrui Liu received the BA degree in 2017 at Shanxi University and the MA degree in 2020 at North University of China. He is currently pursuing a Ph.D. degree in Beijing Jiaotong University, China. His research interest is privacy enhancement technology for cybersecurity applications.

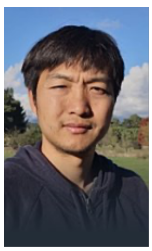
Wei Wang is a full Professor with school of computer and information technology, Beijing Jiaotong University, China. He received the Ph.D. degree from Xi'an Jiaotong University, in 2006. He was a Post-Doctoral Researcher with the University of Trento, Italy, from 2005 to 2006. He was a Post-Doctoral Researcher with TELECOM Bretagne and with INRIA, France, from 2007 to 2008. He was also a European ERCIM Fellow with the Norwegian University of Science and Technology (NTNU), Norway, and with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, from 2009 to 2011. He has authored or co-authored over 100 peer-reviewed articles in various journals and international conferences. His recent research interests lie in data security and privacy-preserving computation. He is an Elsevier "highly cited Chinese Researchers". He is an Editorial Board Member of Computers & Security and a Young AE of Frontiers of Computer Science.

Hong-Liang Ma is currently an associate professor of Shihezi University. He received the Ph.D. degree from the Beijing Jiaotong University in 2018. His research interests mainly include artificial intelligence security, Web security, and anomaly detection.



Bin Wang is a professor of Zhejiang Key Laboratory of Multi-dimensional Perception Technology, Application and Cybersecurity and Zhejiang University. He received the Ph.D. degree from the China National Digital Switching System Engineering & Technological R&D Center in 2010. His research interests mainly include Internet of Things security, cryptography, artificial intelligence security, and new network security architecture.

Zhen Han is currently a Professor at the School of Computer and Information Technology of Beijing Jiaotong University. He received his Ph.D. degree from China Academy of Engineering Physics, in 1991. He has authored or co-authored over 100 papers in various journals and international conference. His main research interests are information security architecture and trusted computing.



Yufei Han is a senior researcher of INRIA France. His research interests include trustworthy and interpretable AI technologies for cybersecurity applications, as well as the adversarial robustness of AI systems. He has published over 50 research papers on top-tiered venues on AI and cybersecurity, like IEEE S&P, CCS, KDD, NDSS and AAAI. He regularly serves as program committees and peer reviews in these venues.

TABLE 6
The architecture of generator in CGIR attack for MNIST and CIFAR10.

Layer	MNIST				CIFAR10 / CIFAR100			
	Filter/Stride	Resample	BN	Output Size	Filter/Stride	Resample	BN	Output Size
Linear/View	-	-	-	128*7*7	-	-	-	128*4*4
Conv/GLU	3*3/1	Up	Y	64*14*14	3*3/1	Up	Y	64*8*8
Conv/GLU	3*3/1	Up	Y	32*28*28	3*3/1	Up	Y	32*16*16
Conv/Softmax	3*3/1	-	-	1*28*28	3*3/1	UP	Y	3*32*32

APPENDIX A

The architectures of generator for MNIST, CIFAR10 and CIFAR100 are provided in Table 6, where 'Y' indicates the layer is followed by a BN layer. The architecture of the generator for CelebA-HQ and ImageNet datasets is similar to that of CIFAR10 and CIFAR100, except that the number of upsampling blocks is increased accordingly.

APPENDIX B

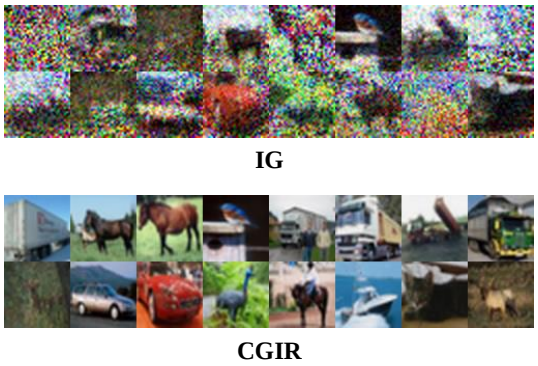


Fig. 12. Comparison results on CIFAR10 with batch size 16.

Fig. 13. CGIR attack on CIFAR10 with batch size 64.

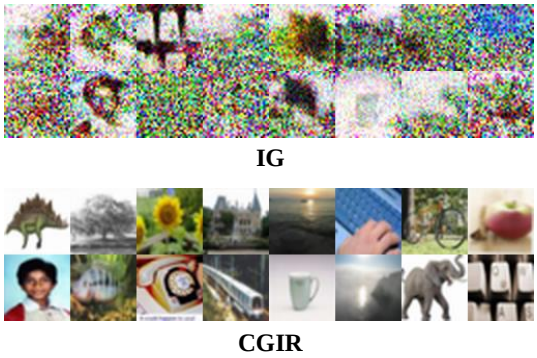


Fig. 14. Comparison results on CIFAR100 with batch size 16.

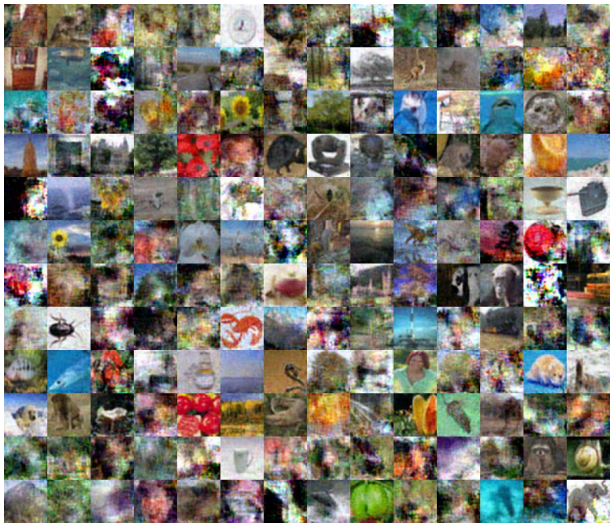


Fig. 15. CGIR attack on CIFAR100 with batch size 168.