High-Fidelity Gradient Inversion in Distributed Learning

Zipeng Ye[1, 2], **Wenjian Luo**[1, 2, 3*], **Qi Zhou**[1, 2], **Yubo Tang**[1, 2]

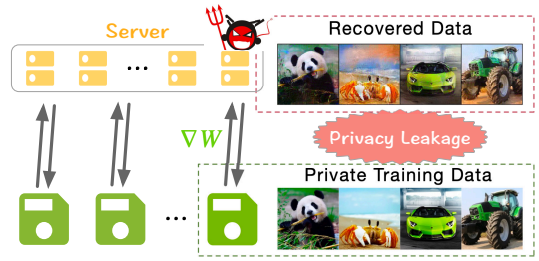
[1]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen
[2]Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
[3]Peng Cheng Laboratory
22B351009@stu.hit.edu.cn, luowenjian@hit.edu.cn, {22S051036, 22S151061}@stu.hit.edu.cn

Abstract

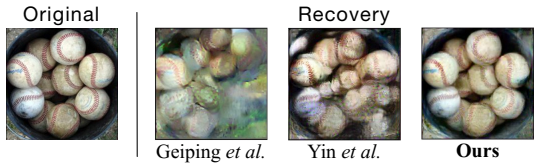
Distributed learning frameworks aim to train global models by sharing gradients among clients while preserving the data privacy of each individual client. However, extensive research has demonstrated that these learning frameworks do not absolutely ensure the privacy, as training data can be reconstructed from shared gradients. Nevertheless, the existing privacy-breaking attack methods have certain limitations. Some are applicable only to small models, while others can only recover images in small batch size and low resolutions, or with low fidelity. Furthermore, when there are some data with the same label in a training batch, existing attack methods usually perform poorly. In this work, we successfully address the limitations of existing attacks by two steps. Firstly, we model the coefficient of variation (CV) of features and design an evolutionary algorithm based on the minimum CV to accurately reconstruct the labels of all training data. After that, we propose a stepwise gradient inversion attack, which dynamically adapts the objective function, thereby effectively and rationally promoting the convergence of attack results towards an optimal solution. With these two steps, our method is able to recover high resolution images (224×224 pixel, from ImageNet and Web) with high fidelity in distributed learning scenarios involving complex models and larger batch size. Experiments demonstrate the superiority of our approach, reveal the potential vulnerabilities of the distributed learning paradigm, and emphasize the necessity of developing more secure mechanisms. *Source code is available at https://github.com/MiLab-HITSZ/2023YeHFGradInv.*

Introduction

It is a consensus view that using more training data can help improve the performance of AI models (Pati et al. 2022). However, data is usually distributed among different users, organizations and institutions, and the behaviors of sharing data for training model collaboratively will obviously result in privacy leakage (Zhang et al. 2022b). Distributed learning is an emerging paradigm for training AI models collaboratively without the explicit sharing of data (Li et al. 2020; Banabilah et al. 2022; Abreha, Hayajneh, and Serhani 2022). One basic setting of distributed learning is the "client-server" architecture, in which each client trains locally using its own data and uploads the gradients to the server (McMahan et al. 2017; Karimireddy et al. 2020; Long et al. 2020). The server aggregates these uploaded gradients from participants and distributes the updated model back to them.



(a) Overview of threat model in distributed learning



(b) Qualitative comparison between prior art and ours.

Figure 1: Privacy leakage in distributed learning. Client sends the gradients $\nabla \mathbf{W}$ averaged from the batch training images to the server. The honest-but-curious server recovers images from $\nabla \mathbf{W}$ via gradient inversion.

Although such an architecture has no need to share training data across clients and server, Zhu et al. (Zhu, Liu, and Han 2019) have demonstrated that it is still possible for honest-but-curious server to reconstruct client-side training data from shared gradients. Specifically, they have recovered training data from randomly initialized noise by gradient matching, which aims to minimize the Euclidean distance between ground-truth gradients and generative gradients of the recovered data. The feasibility of their method (a.k.a., DLG) is verified in a distributed learning scenario, where pixel-level image reconstruction is achieved.

However, there are still some limitations of DLG. First, DLG only works well on reconstructing low-resolution training data in shallow models. Moreover, in practical distributed learning scenario, clients usually input a whole batch of data at one iteration for training, then upload the batch-averaged gradients, rather than the gradients of single

*Corresponding Author

image. Such an averaging significantly increases the difficulty of gradient inversion attacks (GIAs), and even leads to the failure of the convergence of the optimization process. Recent work has shown that recovering the training data from gradients with high fidelity and high quality is still a challenging problem (Geiping et al. 2020). Inspired by DeepInversion (Yin et al. 2020), Yin et al. (Yin et al. 2021) have proposed a novel gradient inversion method, which utilizes the information stored in the batch normalization layers to help synthesize realistic images. However, their method suffers from a significant drawback, i.e., when there are identically labeled data in the batch, their method fails to accurately reconstruct the labels of these data, resulting in a significant decrease in attack performance.

To address the limitations of existing methods, this paper presents a novel approach. Firstly, we model the coefficient of variation of data features and propose an evolutionary algorithm based on the minimum coefficient of variation. This algorithm enables rapid and precise label reconstruction of the training data. Additionally, leveraging the characteristics of gradient backpropagation, we progressively introduce the gradients that require matching, transforming the optimization problem into a step-by-step process from simplicity to complexity. Moreover, we devise a loss scheduling strategy that partitions the gradient inversion into two distinct stages: content recovery and quality improvement. We also elegantly solve the problem of content offset among the recovered images by gradient dropout, without the multi-seed computation like (Yin et al. 2021). The advantageousness of these designs are demonstrated in our experiments.

Our main contributions are summarized as follows.

- We design an evolutionary strategy based on the CV of the features, which enables rapid and accurate recovery of data labels, thus significantly improving the performance of gradient inversion attack.
- We propose a stepwise gradient inversion attack that, combined with our loss scheduling strategy, achieves higher fidelity recovery of private training data at large batch size compared to prior arts. And we elegantly solve the spacial offset problem of the recovered images by gradient dropout, which can be easily implemented.
- We also visually show that gradients from different layers will lead to different degrees of privacy leakage, which may inspire the communication mechanism design for future distributed learning.

Related Works

Privacy leakage in distributed learning. Zhu et al. (Zhu, Liu, and Han 2019) have firstly proposed to reconstruct clients' data and labels simultaneously using only gradients in distributed learning. However, their approach is only feasible when the model, batch size and the resolution are small. (Zhao, Mopuri, and Bilen 2020) found that the ground-truth labels can be directly recovered by the sharing gradients, without any optimization operation. So they divided the process of gradient inversion into two steps: label restoration and data recovery, which facilitated the convergence of the optimization process. But the limitation is that their method can only infer the label of a single training sample and is not applicable to the batch training data.

After that, (Geiping et al. 2020) introduced image prior regularization to reconstruct images with high resolution. But their method only works well at a small batch size. (Yin et al. 2020) have firstly proposed to recover high fidelity images with the information stored in the batch normalization layers. This method was transplanted by (Yin et al. 2021) to the gradient inversion domain and achieved high quality images recovery. And for solving the spacial offset problem, (Yin et al. 2021) have utilized multi-seeds optimization and RANSAC-flow image alignment (Shen et al. 2020), which requires more computing resources. Furthermore, their method can only accurately recover the labels when all the training data within the batch possess different labels. Otherwise, the effectiveness of their approaches will be compromised.

There are also some linear equation solving-based GIAs (Trieu et al. 2017; Zhu and Blaschko 2021; Chen and Campbell 2021), where the adversaries aim to establish linear equations that capture the relationship between the input and gradients, then leveraging linear equation solver to determine the input. However, these methods usually assume that the target model is linear and they are limited on reconstructing data from batch averaged gradients.

Potential countermeasures. An intuitive defense approach is to degrade the shared gradients, which is commonly achieved through two methods: gradient sparsification (Zhu, Liu, and Han 2019; Li et al. 2022) and gradient perturbation (Abadi et al. 2016; Wei and Liu 2021). However, empirical results demonstrate that gradient sparsification fails to provide effective privacy protection, even when the pruning ratio is set as high as 90% (Huang et al. 2021, 2020). Regarding gradient perturbation, differential privacy techniques (Dwork 2006) are commonly employed to provide privacy guarantees. However, achieving strict privacy guarantees necessitates the addition of excessively large noise to the gradients, thereby significantly impacting model utility.

In terms of cryptography-based defense methods, secure multi-party computation (SMPC) (Zhang et al. 2022a; Mugunthan et al. 2019) and homomorphic encryption (Zhang et al. 2020; Jia et al. 2021) are two frequently utilized approaches. SMPC (Yao 1982) ensures that individual clients receive accurate computation results while preventing them from acquiring any additional information beyond the results. And homomorphic encryption (Rivest et al. 1978) aggregates the gradients uploaded by clients in the form of ciphertexts, guaranteeing that an attacker cannot derive any private information from the encrypted shared gradients. Although cryptography-based techniques do not compromise the model utility, they considerably increase computation time and necessitate more communication bandwidth, thereby posing significant implementation challenges.

Methodology

In this section, we will first introduce the threat model. After that, we formulate the problem mathematically. Finally, we give the implementation details of our approach.

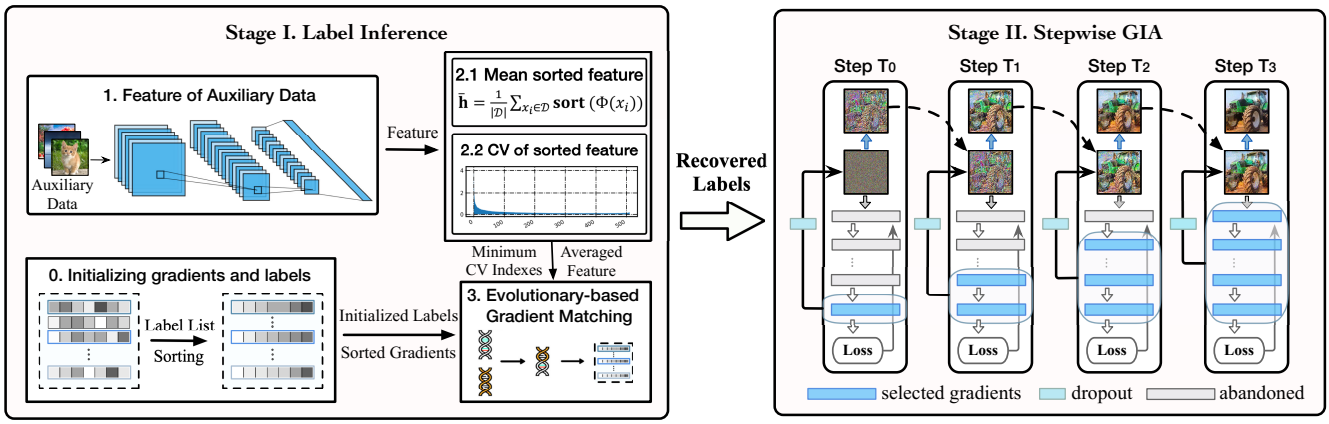


Figure 2: Pipeline of the proposed attack method. At the initial of Stage I, we coarsely infer the labels through shared gradients. Subsequently, leveraging the sorted feature of auxiliary data, the averaged sorted feature as well as the index associated with the minimum CV are calculated. These results are then utilized together to accurately infer the ground-truth labels of training data using an evolutionary algorithm. After that, in Stage II, we reconstruct the privacy data in a stepwise manner by gradually introducing the gradients that need to be matched.

Threat Model

In gradient inversion attack, the adversary is considered to be an honest-but-curious server. In fact, the scenario under this assumption is a centralized distributed learning framework, which has a central parameter server. Another more general scenario is that in a decentralized distributed learning framework, the adversary can be any individual involved in the training of the global model. In either case, the gradient inversion attack can be successfully implemented since adversaries are both able to obtain the gradients from other clients. The overview of threat model is shown in Fig. 1, where the curious server receives the gradients from an honest client and maliciously, leverages an elaborate inversion algorithm to recover the privacy data.

Problem Formulation

Given a model f_W, and a batch of training data pairs $\bigcup_i^N \{(x_i, y_i)\}$ with image x_i and label y_i, the batch-averaged ground-truth gradients $\nabla \mathbf{W}$ can be calculated as $\nabla \mathbf{W} = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\mathbf{W}} \ell(f_W(x_i), y_i)$, where ℓ is the loss function. And assuming $(\mathbf{x}', \mathbf{y}') = \bigcup_i^N \{(x_i', y_i')\}$ as the recovered data batch, similarly, we feed them into the model f_W and get generative gradients $\nabla \mathbf{W}' = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\mathbf{W}} \ell(f_W(x_i'), y_i')$. The purpose of the curious server is to find the optimal minibatch $(\mathbf{x}^*, \mathbf{y}^*) = \bigcup_i^N \{(x_i^*, y_i^*)\}$, which satisfies:

$$(\mathbf{x}^*, \mathbf{y}^*) = \underset{(\mathbf{x}', \mathbf{y}')}{\arg\min}\, \ell_{grad}(\nabla \mathbf{W}', \nabla \mathbf{W}) + \mathcal{R}_{image}(\mathbf{x}'), \quad (1)$$

where ℓ_{grad} measures the distance from $\nabla \mathbf{W}'$ to $\nabla \mathbf{W}$, and $\mathcal{R}_{image}$ promotes the recovered images to be realistic.

Stage I. Label Inference

Simultaneously optimizing the images $\mathbf{x}'$ as well as the labels $\mathbf{y}'$, the result usually does not converge to the reasonable solution (obtained images are often close to noise). Zhao et al. (Zhao, Mopuri, and Bilen 2020) found that single label can be directly recovered from the gradients of the final fully connected layer $W^{FC} \in R^{n \times m}$ (where n is the total number of categories and m is the input dimension of the last fully connected layer), thus significantly improving the performance of GIAs. However, existing methods cannot accurately reconstruct the labels of all data in a batch when these labels are not unique (i.e., there are different data in the batch with the same label). Therefore, in the first stage of our method, we aim to design an evolutionary-based attack for accurately reconstructing the labels, and the pipeline is shown in Fig. 2. Note that we assume that the number of training data N utilized by the victim is known. This assumption is used in all related studies, as it is argued to be reasonable due to the requirement of data amount sharing in benchmark aggregation algorithms. Moreover, from a privacy standpoint, we consider this assumption to be reasonable as it can be potentially breached by an attacker through exhaustive search or brute-force methods.

Label Initializing Considering a single training data with label k, the gradient of the cross-entropy loss on the logits at index i (the input of SoftMax layer) can be mathematically represented as (Zhao, Mopuri, and Bilen 2020):

$$g_i = \begin{cases} p_i - 1, & \text{if } i = k \\ p_i, & \text{if } i \in \{1, 2, \cdots, n\} \backslash \{k\} \end{cases} \quad (2)$$

where $p_i > 0$ is the i-th output of SoftMax layer.

In general, logits is outputted by a fully connected layer $\mathbf{W}^{FC}$, and the input $\mathbf{h} \in \mathbb{R}^m$ of this fully connected layer is usually activated by a non-negative function, such as ReLU or Sigmoid (i.e., the entries of $\mathbf{h}$ are all non-negative). Since the i-th row gradients of $\mathbf{W}^{FC}$ can be calculated as $\nabla \mathbf{W}^{FC}_{i,:} = g_i \mathbf{h}^\intercal$. So we can get for $\forall j$, it satisfies:

$$\begin{cases} \nabla \mathbf{W}^{FC}_{ij} \leqslant 0, & \text{if } i = k \\ \nabla \mathbf{W}^{FC}_{ij} \geqslant 0, & \text{if } i \in \{1, 2, \cdots, n\} \backslash \{k\} \end{cases} \quad (3)$$

Therefore, the ground-truth label k can be directly inferred from the row index where the signs of gradients are all non-postive. For neural networks, in the early stage of training, it is easy to derive that p_i in (2) is close to 0, which indicates that $|p_i - 1| \gg p_i$. So with the batch training data, where $\nabla \mathbf{W}^{FC}_{i,:}$ is averaged from the whole batch, the coarse inference of labels in the batch can be expressed as $\hat{\mathbf{y}} = \bigcup_i \{y_i \mid \text{Mean}(\nabla \mathbf{W}^{FC}_{y_i,:}) < 0\}$. However, it is obvious that the number of training data in the batch belonging to each label is still unknown.

Feature approximating Denoting $\mathbf{k}$ as the indexes which are obtained from the coarse labels in $\hat{\mathbf{y}}$ (i.e., $\mathbf{k}$ consists of the labels in $\hat{\mathbf{y}}$), then by the significant magnitude difference between $p_i - 1$ and p_i, we can set the gradient with large values in $\nabla \mathbf{W}^{FC}$ as $\nabla \mathbf{W}^{FC}_{\mathbf{k},:}$. Our method is motivated by the assumption that there is a mean feature $\bar{\mathbf{h}}$, which is similar to all data's features (actually it is impossible). Then based on the mean feature $\bar{\mathbf{h}}$, we can optimize the target vector $\mathbf{v} \in \mathbb{R}^{|\hat{\mathbf{y}}|}$ (where each dimension of $\mathbf{v}$ represents the number of data belong to each label in $\hat{\mathbf{y}}$) with the objective function:

$$\arg\min_{\mathbf{v}} \mathcal{J}(\mathbf{v}) = 1 - \frac{\left\langle -1 \cdot \mathbf{v}\bar{\mathbf{h}}^{\intercal}, \nabla \mathbf{W}^{FC}_{\mathbf{k},:} \right\rangle}{\left\| \mathbf{v}\bar{\mathbf{h}}^{\intercal} \right\| \left\| \nabla \mathbf{W}^{FC}_{\mathbf{k},:} \right\|}, \tag{4}$$

note that we set $p - 1$ as -1 directly in (4), since they are approximately equal at the beginning of training. After optimizing for (4), combined with the total amount of training data N (note that $\mathbf{v}$ in (4) is scale-independent, so we need to normalize $\mathbf{v}$ using N), we are able to infer all the labels.

However, different training data exhibit different feature activations, and there is no single mean feature $\bar{\mathbf{h}}$ that can effectively represent the feature activations of all the data. To address this challenge, we propose an approximating method based on the coefficient of variation of features. Firstly, we input publicly available auxiliary data $x_i \in \mathcal{D}$ to get the feature $\mathbf{h}_i = \Phi(x_i)$ (where Φ is the part of the model used to extract feature), then we sort $\mathbf{h}_i$ into $\mathbf{h}^S_i = \text{sort}(\mathbf{h}_i)$. Subsequently, we define the notation $\mathbf{t}$ as the indexes of top-M (in our experiment M = 20) smallest CV of $\mathbf{h}^S_i$:

$$\mathbf{t} = \arg\text{sort} \left\{ \underbrace{\text{Std}_{x_i}(\mathbf{h}^S_i) \cdot \mathbb{E}_{x_i}(\mathbf{h}^S_i)^{-1}}_{\text{CV}} \right\} [0 : \text{M}], \tag{5}$$

which indicates that the entries of various data's $\mathbf{h}^S_i$ exhibit minimal differences at indexes $\mathbf{t}$, and the CV calculated by the features of auxiliary data are shown in Fig. 3. Now, our final optimization objective can be expressed as:

$$\arg\min_{\mathbf{v}} \mathcal{J}(\mathbf{v}) = 1 - \frac{\left\langle -\mathbf{v}\bar{\mathbf{h}}^{\intercal}_{\mathbf{t}}, \text{sort}(\nabla \mathbf{W}^{FC}_{\mathbf{k},:})[:, \mathbf{t}] \right\rangle}{\left\| \mathbf{v}\bar{\mathbf{h}}^{\intercal}_{\mathbf{t}} \right\| \left\| \text{sort}(\nabla \mathbf{W}^{FC}_{\mathbf{k},:})[:, \mathbf{t}] \right\|}, \tag{6}$$

where $\bar{\mathbf{h}} = \mathbb{E}_{x_i}(\mathbf{h}^S_i)$, $\nabla \mathbf{W}^{FC}$ is the shared gradients, $\mathbf{k}$ represents the coarse labels, $\mathbf{t}$ and $\bar{\mathbf{h}}$ are obtained by auxiliary data, thus the only unknown term is the vector $\mathbf{v}$ which needs to be optimized.

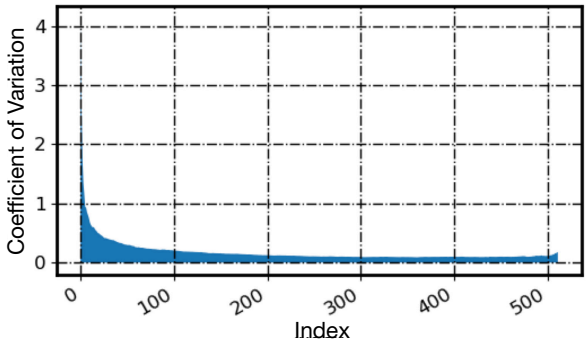


Figure 3: The coefficient of variation for the sorted features exhibits a distinct trend. As the feature index increases, the CV initially decreases and subsequently increases.

Evolutionary matching For optimizing $\mathbf{v}$, we customize the evolutionary algorithm as follows.

(1) Initializing Population: At first we define a rescaling function as $\text{Rs}(\cdot) = \Phi_3 \circ \Phi_2 \circ \Phi_1(\cdot)$, where Φ_1 normalizes the input vector to a magnitude of N, Φ_2 modifies the each entry $\mathbf{v}_i$ to $\max\{\text{Floor}(\mathbf{v}_i), 1\}$, and Φ_3 adjusts the last entry of $\mathbf{v}$ to achieve $\|\mathbf{v}\|_1 = N$. For the population P, we randomly initialize m ($m = 20$ in our experiment) group of vectors and map them by rescaling function Rs to $\mathbf{v}^{(i)}$. Then we have the initial population $P = \{\mathbf{v}^{(i)}\}^m_{i=1}$.

(2) Crossover: For each parent individual $\mathbf{v}^{(i)}$, we randomly select another parent $\mathbf{v}^{(j)}$ from P, then the child $\mathbf{w}^{(i)}$ can be expressed as:

$$\mathbf{w}^{(i)}_k = \begin{cases} \mathbf{v}^{(i)}_k, & \text{if } \gamma_k < p_r \\ \mathbf{v}^{(j)}_k, & \text{if } \gamma_k \geq p_r \end{cases} \tag{7}$$

where $(\cdot)_k$ represents the k^{th} entry of vector $(\cdot)$, γ_k is sampled from the uniform distribution $U(0, 1)$, and p_r is the preset crossover threshold (0.5 in this paper). And the final child can be calculated by $\mathbf{w}^{(i)} = \text{Rs}(\mathbf{w}^{(i)})$.

(3) Mutation: For each child $\mathbf{w}^{(i)}$, we randomly choose one entry greater than 1 to subtract 1 from it, and randomly choose another entry to plus 1.

(4) Selection: For each pair of parent $\mathbf{v}^{(i)}$ and child $\mathbf{w}^{(i)}$, we construct new population P^+ with objective (4):

$$P^+ = \bigcup_i \left\{ \mathbf{v}^{*(i)} | \mathbf{v}^{*(i)} = \arg\min_{\mathbf{v} \in \{\mathbf{v}^{(i)}, \mathbf{w}^{(i)}\}} \mathcal{J}(\mathbf{v}) \right\}. \tag{8}$$

By repeating crossover, mutation and selection, labels of all training data can be rapidly (requires fewer than 200 iterations and takes less than 1 second) and accurately inferred. Pseudocode of this part is provided in Appendix A.

Stage II. Stepwise GIA

Stepwise optimization objective In Stage II, we aim to reconstruct the training data using the labels already inferred in Stage I. Nevertheless, achieving high-fidelity reconstruction of batch images in larger models is typically challenging

due to the inherent difficulty of finding the optimal solution, which locates in natural image manifold while its gradients simultaneously matching the large number of ground-truth gradients. Actually, the result of such a gradient matching optimization problem tends to fall into a local optimum (see the result of Deep Gradient Leakage in Fig. 5). Therefore, we propose an effective method, which gradually introduces the ground-truth gradients that need to be matched, thus progressively guiding the generative gradients of the recovered images to approximate the ground-truth. Our idea further relaxes the constraint on the optimization and enables the recovery of private data in deeper networks. Without loss of generality, we re-express the logits of network as follows:

$$\xi = \mathbf{W}^l \, \sigma^{l-1} \overbrace{\left(\mathbf{W}^{l-1} \underbrace{\sigma^{l-2}(\psi(\mathbf{x}))}_{\mathbf{h}_{l-2}} + \mathbf{b}^{l-1} \right)}^{\mathbf{h}_{l-1}} + \mathbf{b}^l, \quad (9)$$

where l denotes the l-th layer, $\mathbf{W}$ denotes the weights, σ denotes the activation function, $\mathbf{b}$ denotes the bias and ψ is the mapping function for the first $l-2$ layers. By observing (9), we can infer that the gradient of $\mathbf{W}^l$ is only related to the middle output $\mathbf{h}_{l-1}$ and not to the output of the layer before it. And so is the relation between the gradient of $\mathbf{W}^{l-1}$ and the middle output $\mathbf{h}_{l-2}$. Therefore, middle output can be estimated from corresponding gradients without the need of considering earlier gradients and earlier middle outputs. Subsequently, by incrementally introducing the gradients to be matched one by one, we can drive the output of middle layers from deep to shallow, up to the input data, to converge to the optimum in a stepwise manner (see Fig. 2).

Further, as mentioned in (Yin et al. 2021), gradient inversion is prone to causing a content spacial offset of the recovered image. To solve this problem, they have proposed group consistency regularization, which enhances the recovery quality by multi-seed optimization and image registration, which consumes large amounts of computing resources. We solve this problem smoothly by only performing a random dropout to the gradients that need to be matched. Our experiment results demonstrate the superiority of such an operation. Since labels are already inferred in the first stage, our stepwise optimization objective function can be re-written from (1) to:

$$\mathbf{x}_t^* = \arg\min_{\mathbf{x}'} \underbrace{\left(1 - \frac{\left\langle \nabla\widetilde{\mathbf{W}}_t', \nabla\widetilde{\mathbf{W}}_t \right\rangle}{\left\| \nabla\widetilde{\mathbf{W}}_t' \right\| \left\| \nabla\widetilde{\mathbf{W}}_t \right\|} \right)}_{\tilde{\ell}_{grad}} + \mathcal{R}_{image}(\mathbf{x}'), \quad (10)$$

where t represents different stage, $\nabla\widetilde{\mathbf{W}}_t'$ and $\nabla\widetilde{\mathbf{W}}_t$ are generative gradients and ground-truth gradients respectively, which are both selected at t with dropout.

Image Prior Regularization Total Variation (TV) $\mathcal{R}_{tv}$ is a common used image prior regularizer, which encourages the recovered images to be more constant and coherent. In addition, (Yin et al. 2020) proposes feature distribution regularization $\mathcal{R}_{BN}$, which uses the statistical information of the real image data stored in the batch normalization layers, to promote the realism of the recovered image. These two regularizers can be expressed as:

$$\begin{cases} \mathcal{R}_{tv}(\mathbf{x}') = \sum_{i,j} \left((x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2 \right)^{\frac{\beta}{2}}, \\ \mathcal{R}_{BN}(\mathbf{x}') = \sum_l \left\| \mu_l(\mathbf{x}') - \bar{\mu}_l \right\|_2 + \sum_l \left\| \sigma_l^2(\mathbf{x}') - \bar{\sigma}_l^2 \right\|_2, \end{cases} \quad (11)$$

where β is preset hyperparameter, $\bar{\mu}_l$ and $\bar{\sigma}_l^2$ are batch-wise mean and variance of the natural images in the l^{th} layer of a pre-trained model. And $\mu_l(\mathbf{x}')$ and $\sigma_l^2(\mathbf{x}')$ are batch-wise mean and variance of $\mathbf{x}'$ in the l^{th} layer. The regularizer $\mathcal{R}_{image}$ in (10) is composed by $\mathcal{R}_{image}(\mathbf{x}') = \lambda_{tv}(t)\mathcal{R}_{tv}(\mathbf{x}') + \lambda_{BN}(t)\mathcal{R}_{BN}(\mathbf{x}')$.

A reasonable reconstruction sequence should be to first recover the content of the image, and then improve the quality of the image. Focusing too much on image quality in the early stages will make the optimization more difficult. So in the early stages of our method, we aim to recover the rich-content but low-quality images only by $\tilde{\ell}_{grad}$ term of (10) without $\mathcal{R}_{image}$. In the middle and late stages, the low-quality images are then optimized on the visual level by introducing $\mathcal{R}_{image}$. In this paper, we first employ $\mathcal{R}_{tv}$ to enhance the image quality, and then $\mathcal{R}_{BN}$ to enhance the image realism. This strategy of introducing image priors sequentially is proven to be effective in our experiments. The scheduling strategy for different image priors is as $\lambda_{tv}(t) = \mathbb{I}_{t>T_1}\hat{\lambda}_{tv}$ and $\lambda_{BN}(t) = \mathbb{I}_{t>T_2}\hat{\lambda}_{BN}$, where $\mathbb{I}$ is the indicator function, T_1, T_2 are the steps for introducing corresponding regularizers (assume that the total iteration steps is T), and $\hat{\lambda}_{tv}, \hat{\lambda}_{BN}$ are hyperparameters.

The Complete Iteration In summary, the iterations of the images recovery in distributed learning can be expressed as:

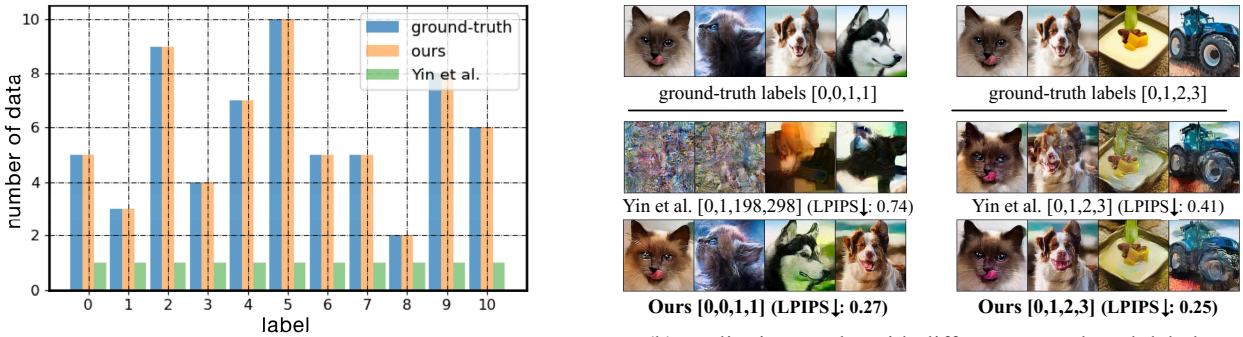
$$\begin{aligned} g_t &\leftarrow \nabla_{\mathbf{x}'_{t-1}} \left(\tilde{\ell}_{grad} + \mathcal{R}_{image}(\mathbf{x}'_{t-1}) \right) \\ \mathbf{x}'_t &\leftarrow \text{Truncate}(\mathbf{x}'_{t-1} - \eta(t) \cdot \text{Sign}(g_t)), \end{aligned} \quad (12)$$

where we use the step learning rate $\eta(t)$ to recover the images in our experiments, $\text{Sign}(\cdot)$ is the sign function, and $\text{Truncate}(\cdot)$ is used to limit the pixels in a natural range (i.e., limit the bound of pixel values).

Experiments

We conduct experiments for large-scale image classification task using ImageNet ILSVRC 2012 dataset (Deng et al. 2009), as well as randomly collected images from Web. Due to page constraints, we put additional results in Appendix B.

Setup and Evaluation Metrics To ensure that the auxiliary data used in Stage I are publicly available, we randomly collected them from Web, which contains 40 images with 224×224 px (we will share them after accepted). Our experiments are implemented on pre-trained ResNet34 (He et al. 2016), which is provided by PyTorch library. And more attack results for models with different structures are given in Appendix B. We use Adam (Kingma and Ba 2014) for optimization with a step learning rate decay, and each batch



(a) results of label inference

(b) qualitative results with different ground-truth labels

Figure 4: Results of label inference and its impact on training data reconstruction. In (a), our method can reconstruct the labels of all training data accurately. And (b) qualitatively shows the attack results in scenarios with different label inference accuracy.

is optimized with 15K iterations on NVIDIA TITAN RTX GPUs. The dropout rate we used in (10) is set as 0.3. We set $\beta = 2$ in (11), $\hat{\lambda}_{tv} = 0.01, \hat{\lambda}_{BN} = 10^{-4}, T_1 = 3,000, T_2 = 5,000$ in scheduling strategy. Note that we have tuned all hyperparameters, including those for the comparative experiments, to the optimum. For the evaluation, we use both the qualititative and quantitative metrics to evaluate the performance of our method, including: (i) Visual Comparison, (ii) Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al. 2018), (iii) Peak Signal-to-Noise Ratio (PSNR) and (iv) Structural Similarity (SSIM) (Wang et al. 2004).

Label Inference We randomly select 64 images from the validation set of ImageNet, and their label distributions are shown in Fig. 4 (a). From Fig. 4 (a), existing method performs poorly on label reconstruction, while our method is able to accurately recover labels. And in the left of Fig. 4 (b), since Yin et al. method cannot accurately recover the labels, the attack performance is poor. From the right of Fig. 4 (b), even if all labels are correctly inferred, our stepwise attack method still shows results with higher quality and fidelity. More results can be found in Appendix B.

Method	Evaluation Metrics		
	LPIPS ↓	PSNR ↑	SSIM ↑
Noise $\mathcal{N}(0, I)$	1.39	9.82	0.21
Deep Leakage	1.01	9.31	0.25
Inverting Gradients	0.58	11.99	0.31
GradInversion	0.39	13.22	0.44
Ours	**0.33**	**15.12**	**0.49**

Table 1: Quantitative comparison with SOTA GIAs and the results are averaged over 100 reconstructed images.

Batch Recovery We conduct experiments on images (224×224 px) with batch size 8 and compare our method with prior SOTA both qualitatively and quantitatively. For the sake of comparability, we choose the data with different labels from ImageNet as a batch (same as (Yin et al. 2021)), thus ensuring that all the other methods can also accurately reconstruct the labels. And as shown in Table 1 and Fig. 5, it is hard to recover images with high fidelity only by gradient matching (last row in Fig. 5). And the image prior Total Variation (4th row in Fig. 5) can significantly improve the quality, but the results are still unsatisfactory. As for (Yin et al. 2021), it does outperform the other two methods in terms of the image realism. And our stepwise attack method shows the best performance, both qualitatively and quantitatively. As for the scenario where the batch has training data with the same labels, we can refer to the results in Fig. 4 (b). More experimental results, including attacking on different models, can be found in the Appendix B.

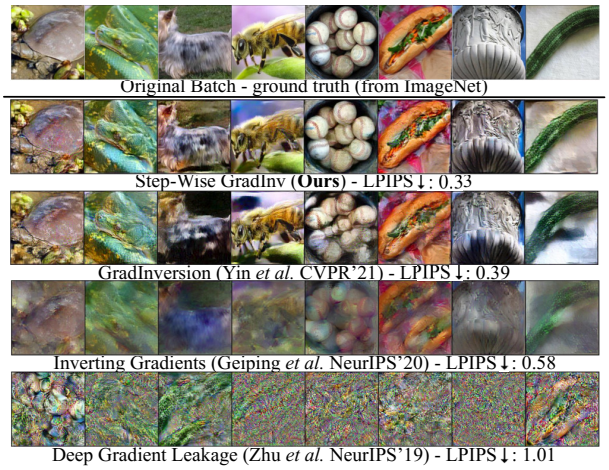


Figure 5: Batch ImageNet images recovery on ResNet34.

Ablation Study In this part, we demonstrate the indispensability of each component used in this paper. The results is shown in Table 2, where ℓ_{grad} means that only cosine similarity is used as gradient matching loss. It can be seen that $\mathcal{R}_{image}$ priors can improve the quality of reconstructed images compared to using ℓ_{grad} only, and when using our stepwise objective $\tilde{\ell}_{grad}$, attack performance can be significantly improved. Loss scheduling strategy can further polish

the recovered images by tuning the recovery process slightly.

Further, in Fig. 6, we have shown the roles of dropout and image priors played in our stepwise GIA. Fig. 6 (a) shows that gradient dropout can effectively mitigate the issue of content offset appearing in reconstructed images. In addition, we give the visual comparison at different iteration stages (i.e., T_1, T_2 and T, see Fig. 6 (b)), so that help readers have a better understanding on each regularization term as well as our loss scheduling strategy.

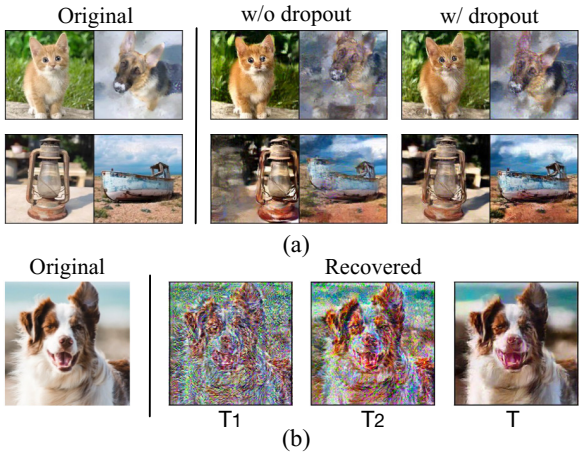


Figure 6: Effect of gradient dropout and image priors: (a) shows the effect of dropout in mitigating content offset, where the images in first row are from ImageNet and the second row are from Web; (b) shows that $\mathcal{R}_{tv}$ can improve the continuity of image pixels (compare T_2 with T_1) and $\mathcal{R}_{BN}$ can further enhance the image realism (compare T with T_2).

Effect of Batch Size Fig. 7 shows the performance of private images recovery on different batch size. We observe that the LPIPS value of the recovered images rises gradually as batch size grows, which indicates that batch size indeed affects the quality of the recovery. However, Fig. 7 also shows that our stepwise gradient inversion, with the setting of batch size 32, is still able to realize a high fidelity reconstruction. More attack results can be found in Appendix B.

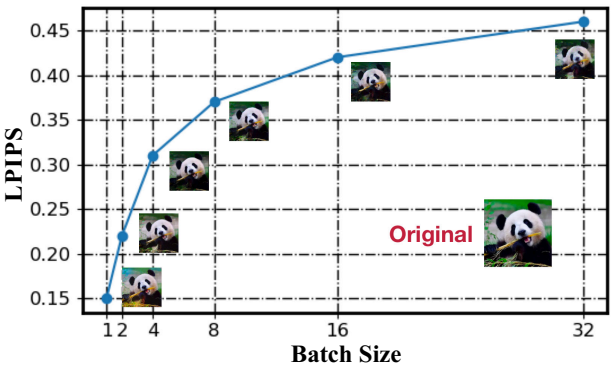


Figure 7: The impact of different batch size on LPIPS value.

Optimization Objective	Evaluation Metrics		
	LPIPS ↓	PSNR ↑	SSIM ↑
Noise $\mathcal{N}(0, I)$	1.39	9.82	0.21
ℓ_{grad}	0.80	8.34	0.27
$+\mathcal{R}_{image}$	0.57	10.79	0.34
$+\tilde{\ell}_{grad}$	0.27	14.71	0.50
+ Schedule	**0.23**	**15.99**	**0.55**

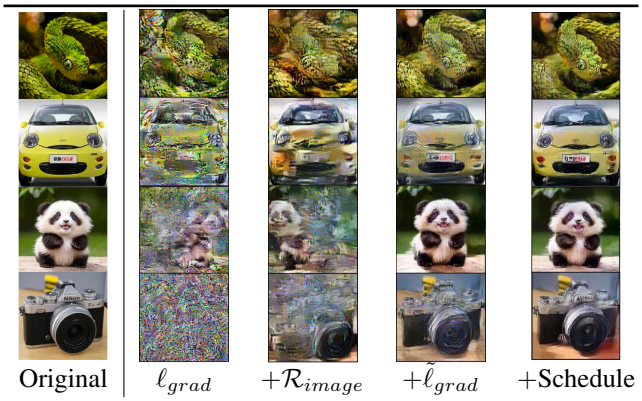


Table 2: The effect of each component on the image recovery, and experiments are implemented with batch size 8.

Using Different Layers Since our method gradually adds gradients from deep to shallow, we aim to further explore the effect of gradients from different layers on the final results. Generally, ResNet consists of 4 blocks and we choose to use different blocks' gradients for image recovery, and the results are shown in Fig. 8. We observe that the deepest block (block 4) owns the richest image feature information, and the training data can be recovered with higher quality using block 4 only. The block 3 and block 2 can supply a small amount of information about the details. However, when the gradients of block 1 is added, the quality of the recovery decreases (so we do not use block 1 in all previous experiments). This is because block 1 is too shallow and the gradients of shallow layers are directly influenced by specific pixel values. However, our method does not care too much about the small color difference between the recovery and the original images (cosine similarity focuses more on direction but not magnitude (Geiping et al. 2020)), which results in the optimization algorithm forcibly patching the recovery to match the gradients of the shallow layers when block 1 is introduced suddenly (thus unnatural regions arise). Also, we find that the quality of the recovery is very poor using only block 2 and 3 without block 4. It means that in distributed learning framework, a large amount of useful information is stored in deeper gradients. Therefore, for alleviating the communication pressure in distributed learning, we can focus more on transferring deeper gradients with some privacy-preserving measures (Shokri and Shmatikov 2015; Bonawitz et al. 2017; Aono et al. 2017).

More Scenarios In addition, we examine the proposed attack in more different scenarios including: employing larger

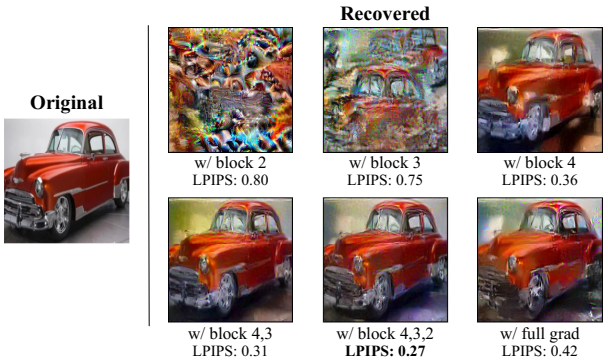


Figure 8: We recover images using gradients from different blocks (4 blocks in total, corresponding to $64, 128, 256, 512$ channels) of ResNet. We use block 1 to denote the block with 64 channels, and block 2 for 128 channels, and so on.

batch sizes, omitting the regularization term $\mathcal{R}_{BN}$, utilizing smaller models, and evaluating the attack's performance under the implementation of defensive strategies. Results are shown in Fig. 9.

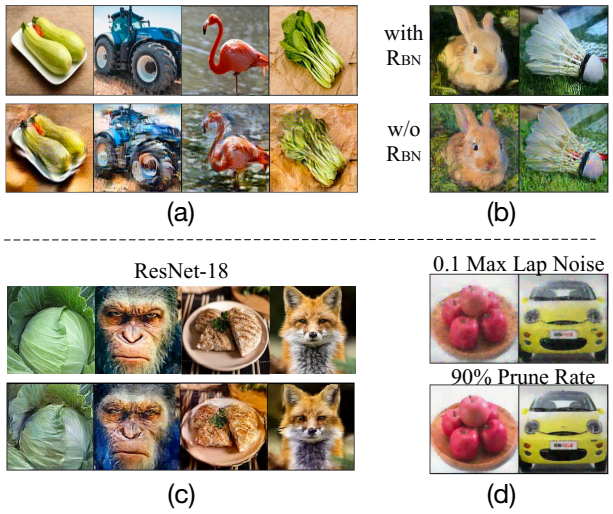


Figure 9: Attack results in more scenarios, where (a) the batch size is 64; (b) different results with and without $\mathcal{R}_{BN}$; (c) with target model ResNet-18; (d) with additive noise and gradient pruning defense, and the noise scale is set as 0.1 times the maximum gradient of each layer.

From Fig. 9, we can see that: (1) obviously, 32 is not the upper limit of our method's capabilities (batch size 64 in Fig. 9(a)); (2) $\mathcal{R}_{BN}$ prior can enhance the reality of reconstructed images, but without $\mathcal{R}_{BN}$, our attack is still sufficient to leak a significant amount of privacy; (3) proposed attack can also achieve satisfactory results with smaller model (ResNet-18 in Fig. 9(c)); (4) simple additive noise and gradient pruning can not effectively resist our attack.

Conclusion

We propose a novel two-stage gradient inversion attack approach, which can realize high fidelity batch images recovery in distributed learning by evolutionary label inference and stepwise gradient inversion. Experiment results show the superiority of our approach and shed light on the potential vulnerability of the distributed learning paradigm. And at the final of this paper, we visually show that gradients from different layers usually own different amounts of data information, which may inspire effective communication mechanism design for future distributed learning.

Acknowledgments

This study is supported by the National Key R&D Program of China (Grant No. 2022YFB3102100), Shenzhen Fundamental Research Program (Grant No. JCYJ20220818102414030), the Major Key Project of PCL (Grant No. PCL2022A03), Shenzhen Science and Technology Program (Grant No. ZDSYS20210623091809029), Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (Grant No. 2022B1212010005).

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 308–318.

Abreha, H. G.; Hayajneh, M.; and Serhani, M. A. 2022. Federated learning in edge computing: a systematic survey. *Sensors*, 22(2): 450.

Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S.; et al. 2017. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5): 1333–1345.

Banabilah, S.; Aloqaily, M.; Alsayed, E.; Malik, N.; and Jararweh, Y. 2022. Federated learning review: Fundamentals, enabling technologies, and future applications. *Information Processing & Management*, 59(6): 103061.

Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 1175–1191.

Chen, C.; and Campbell, N. 2021. Understanding training-data leakage from gradients in neural networks for image classification. In *NeurIPS Workshop*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 248–255.

Dwork, C. 2006. Differential privacy. In *International Conference on Automata, Languages and Programming*.

Geiping, J.; Bauermeister, H.; Dröge, H.; and Moeller, M. 2020. Inverting gradients-how easy is it to break privacy in

federated learning? *Advances in Neural Information Processing Systems*, 33: 16937–16947.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778.

Huang, Y.; Gupta, S.; Song, Z.; Li, K.; and Arora, S. 2021. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems*, 34: 7232–7241.

Huang, Y.; Song, Z.; Li, K.; and Arora, S. 2020. Instahide: Instance-hiding schemes for private distributed learning. In *International Conference on Machine Learning*, 4507–4518.

Jia, B.; Zhang, X.; Liu, J.; Zhang, Y.; Huang, K.; and Liang, Y. 2021. Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT. *IEEE Transactions on Industrial Informatics*, 18(6): 4049–4058.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the International Conference on Machine Learning*, 5132–5143.

Kingma, D. P.; and Ba, J. 2014. ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3): 50–60.

Li, Z.; Zhang, J.; Liu, L.; and Liu, J. 2022. Auditing Privacy Defenses in Federated Learning via Generative Gradient Leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10132–10142.

Long, G.; Tan, Y.; Jiang, J.; and Zhang, C. 2020. Federated learning for open banking. In *Federated Learning: Privacy and Incentive*, 240–254. Springer.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282.

Mugunthan, V.; Polychroniadou, A.; Byrd, D.; and Balch, T. H. 2019. Smpai: Secure multi-party computation for federated learning. In *NeurIPS Workshop*.

Pati, S.; Baid, U.; Edwards, B.; Sheller, M.; Wang, S.-H.; Reina, G. A.; Foley, P.; Gruzdev, A.; Karkada, D.; Davatzikos, C.; et al. 2022. Federated learning enables big data for rare cancer boundary detection. *Nature Communications*, 13(1): 7346.

Rivest, R. L.; Adleman, L.; Dertouzos, M. L.; et al. 1978. On data banks and privacy homomorphisms. *Foundations of Secure Computation*, 4(11): 169–180.

Shen, X.; Darmon, F.; Efros, A. A.; and Aubry, M. 2020. Ransac-flow: generic two-stage image alignment. In *European Conference on Computer Vision*, 618–637.

Shokri, R.; and Shmatikov, V. 2015. Privacy-preserving deep learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 1310–1321.

Trieu, P. L.; Aono, Y.; Hayashi, T.; Wang, L.; and Moriai, S. 2017. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5): 1333–1345.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612.

Wei, W.; and Liu, L. 2021. Gradient leakage attack resilient deep learning. *IEEE Transactions on Information Forensics and Security*, 17: 303–316.

Yao, A. C. 1982. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science*, 160–164. IEEE.

Yin, H.; Mallya, A.; Vahdat, A.; Alvarez, J. M.; Kautz, J.; and Molchanov, P. 2021. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16337–16346.

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8715–8724.

Zhang, C.; Ekanut, S.; Zhen, L.; and Li, Z. 2022a. Augmented multi-party computation against gradient leakage in federated learning. *IEEE Transactions on Big Data*.

Zhang, C.; Li, S.; Xia, J.; Wang, W.; Yan, F.; and Liu, Y. 2020. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *Proceedings of the USENIX Annual Technical Conference*.

Zhang, K.; Song, X.; Zhang, C.; and Yu, S. 2022b. Challenges and future directions of secure federated learning: a survey. *Frontiers of Computer Science*, 16: 1–8.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 586–595.

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

Zhu, J.; and Blaschko, M. 2021. R-gap: Recursive gradient attack on privacy. In *International Conference on Learning Representations*.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32.