

A Survey on Gradient Inversion: Attacks, Defenses and Future Directions

Rui Zhang¹, Song Guo^{1,2*}, Junxiao Wang^{1*}, Xin Xie¹ and Dacheng Tao³

¹ The Hong Kong Polytechnic University

² The Hong Kong Polytechnic University Shenzhen Research Institute

³ JD Explore Academy, JD.com

csrzhang1@comp.polyu.edu.hk, {song.guo, junxiao.wang, xin-ryan.xie}@polyu.edu.hk,
dacheng.tao@gmail.com

Abstract

Recent studies have shown that the training samples can be recovered from gradients, which are called *Gradient Inversion (GradInv)* attacks. However, there remains a lack of extensive surveys covering recent advances and thorough analysis of this issue. In this paper, we present a comprehensive survey on GradInv, aiming to summarize the cutting-edge research and broaden the horizons for different domains. Firstly, we propose a taxonomy of GradInv attacks by characterizing existing attacks into two paradigms: *iteration-* and *recursion-*based attacks. In particular, we dig out some critical ingredients from the iteration-based attacks, including *data initialization*, *model training* and *gradient matching*. Second, we summarize emerging defense strategies against GradInv attacks. We find these approaches focus on three perspectives covering *data obscuration*, *model improvement* and *gradient protection*. Finally, we discuss some promising directions and open problems for further research.

1 Introduction

Distributed learning or federated learning [McMahan *et al.*, 2017; Bonawitz *et al.*, 2019] has become a popular paradigm to achieve collaborative training and data privacy at the same time. In a centralized training process, the parameter server initially sends a global model to each participant. After training with local data, the participants are only required to share gradients for model updates. Then the server aggregates the gradients and transmits the updated model back to each user. However, recent studies have shown that gradient sharing is not as secure as it is supposed to be. We consider an *honest-but-curious* attacker, who can be the centralized server or a neighbor in decentralized training. The attacker can observe gradients of a victim at time t , and he/she attempts to recover data $x(t)$ or labels $y(t)$ from gradients. In general, we name such attacks as *Gradient Inversion (GradInv)* attacks.

A majority of GradInv attacks [Zhu *et al.*, 2019; Geiping *et al.*, 2020; Yin *et al.*, 2021] purpose to minimize the distance between the generated gradients and ground-truth gradients.

In order to generate dummy gradients, a pair of random data and labels are fed to the global model. Taking the distance between the gradients as error and the dummy inputs as parameters, the recovery process can be formulated as an iterative optimization problem. When the optimization procedure converges, the private data is supposed to be fully reconstructed. Moreover, some newly presented studies [Fan *et al.*, 2020; Zhu and Blaschko, 2021; Chen and Campbell, 2021] can also reconstruct the original data in a closed-form algorithm. The key insight is to exploit the implicit relationships among the input data, model parameters and gradients of each layer, and find the optimal solution with minimum error.

To prevent attackers from disclosing privacy from gradient sharing, some cryptography-based methods [Bonawitz *et al.*, 2017; Phong *et al.*, 2018], differential privacy-based approaches [Sun *et al.*, 2021; Wei *et al.*, 2021; Nasr *et al.*, 2021] and pixel-based perturbations [Fan, 2019; Huang *et al.*, 2020] have been proposed to enhance the security and privacy levels. In addition, such privacy leakage can be mitigated by increasing the local iterations or batch sizes [Wei *et al.*, 2020b; Huang *et al.*, 2021] during model training.

However, reviewing the existing surveys of privacy attacks and defenses in distributed learning [Bouacida and Mohapatra, 2021; Jegorova *et al.*, 2021; Wainakh *et al.*, 2021b], we found that there remains a lack of systematic and exhaustive summaries of recent GradInv research. Hence, in this paper, we present a comprehensive survey covering the attacks, defenses and future directions of GradInv. In a nutshell, this paper makes the following key contributions.

- We first propose a novel taxonomy of GradInv attacks. Depending on the optimization objective, we categorize the existing studies into *iteration-based* and *recursion-based* attacks. The former aims at iteratively minimizing the distance between gradients, and the latter recursively recovers the optimal input layer by layer. Specially, we dig out some critical ingredients from the iteration-based attacks, including *data initialization*, *model training* and *gradient matching*. We also analyze the impacts of these ingredients on the attacking effect.
- Then we summarize the emerging and representative defense strategies from the viewpoints of *data obscuration*, *model improvement* and *gradient protection*. In particular, each participant can either obscure original images

*Corresponding Authors

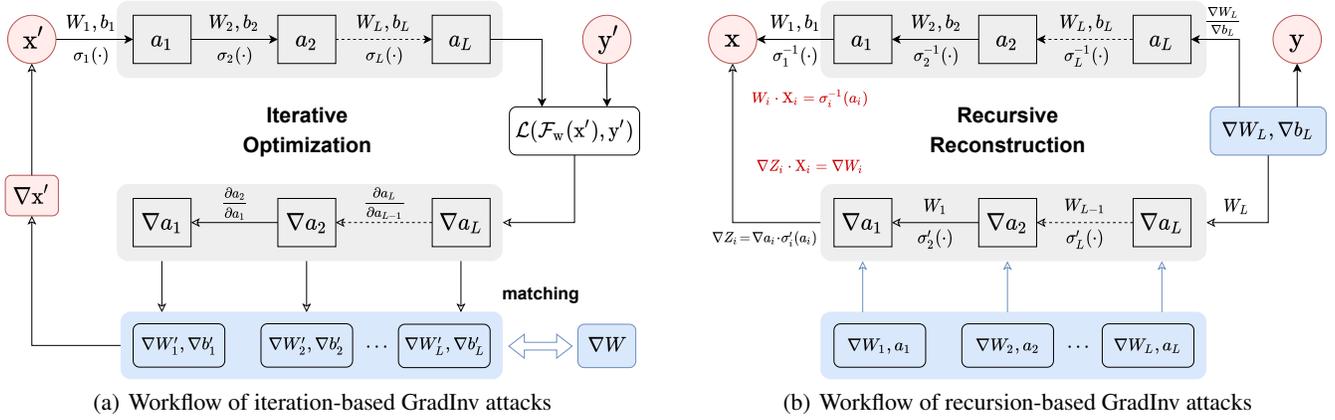


Figure 1: Reconstruction workflow of two types of GradInv attacks. Both procedures include data, model and gradients, which are represented with different colors. **Red**: Reconstructed data, labels or gradients of the dummy inputs. **Gray**: Global model shared in distributed learning, whose parameters and structure are known. **Blue**: Ground-truth gradients and generated gradients for data recovery.)

from a data source, improve the networks from security analysis, or protect the gradients before sharing.

- We finally conclude this survey and discuss some open problems and promising directions, which we hope can shed light on future research. Our proposed directions include the combination of two GradInv attacks, extension to Transformer architecture, and risk quantification between private data and shared gradients.
- To the best of our knowledge, this is the first comprehensive survey focusing on wide-ranging issues of GradInv.

2 Gradient Inversion Attacks

In this section, we provide a taxonomy of GradInv attacks by characterizing the existing works into two paradigms: *iteration* and *recursion*-based attacks. Moreover, we delve into the iteration-based attacks and divide the main components into *data initialization*, *model training* and *gradient matching*. We first describe the workflow of each category and formulate the optimization objectives. Then we introduce the representative studies. A detailed comparison of differences and similarities among recent GradInv attacks is listed in Table 1.

2.1 Iteration-based Data Recovery

In the iteration-based workflow, the attacker first generates a pair of random data \mathbf{x}' and labels \mathbf{y}' , which are regarded as optimization parameters for data recovery. After forward and backward propagation, generated gradients of model weights $\nabla W'$ can be obtained. Here, the bias terms are ignored since they can be integrated into the networks by adding a neuron with the input of 1. According to the distance between generated gradients and ground-truth gradients of a victim, another backward propagation is performed to calculate the gradients of dummy inputs, i.e., $\nabla \mathbf{x}'$ and $\nabla \mathbf{y}'$. From the perspective of reconstructed data, the reconstruction process includes once forward propagation, twice backward propagation and the update of dummy inputs, which is illustrated in Fig. 1(a).

The recovery of private data can be viewed as an iterative optimization process utilizing gradient descent. When the optimization converges, that is, the distance (e.g., ℓ_2 norm) between gradients is close, and the original data is supposed to be fully recovered. The optimization problem is formulated in Eq. (1), where \mathbf{x}^{*} , \mathbf{y}^{*} are the optimized results.

$$\begin{aligned} \mathbf{x}^{*}, \mathbf{y}^{*} &= \arg \min_{\mathbf{x}', \mathbf{y}'} \|\nabla W' - \nabla W\|^2 \\ &= \arg \min_{\mathbf{x}', \mathbf{y}'} \left\| \frac{\partial \mathcal{L}(\mathcal{F}(\mathbf{x}', W), \mathbf{y}')}{\partial W} - \nabla W \right\|^2 \end{aligned} \quad (1)$$

Initialization of Input Data and Labels

To produce dummy gradients and perform gradient matching, the attacker first needs to generate random data and labels. Initialization involves the selection of distribution, the size of original data, and the asynchronous recovery of labels.

Distribution. Random Gaussian noise is most frequently used for data initialization in the majority of GradInv attacks [Zhu *et al.*, 2019; Geiping *et al.*, 2020; Yin *et al.*, 2021]. In addition, constant values [Wang *et al.*, 2020] or random noise sampled from Uniform distribution [Jin *et al.*, 2021] are also presented for data initialization. However, some experiment results demonstrate that GradInv attacks often fail to converge due to bad initialization [Geiping *et al.*, 2020]. Hence, [Wei *et al.*, 2020b] prove that the convergence of a reconstruction process can be guaranteed by Lipschitz continuity:

$$f(\mathbf{x}'^T) - f(\mathbf{x}'^{*T}) \leq \frac{2L \|\mathbf{x}'^0 - \mathbf{x}'^{*0}\|^2}{T} \quad (2)$$

where L is the Lipschitz constant, T denotes the condition of termination, and \mathbf{x}'^0 , \mathbf{x}'^T , \mathbf{x}'^{*} represent the initializing data, terminated results and optimal recovery, respectively. Based on Eq. (2), the authors propose patterned randomization and theoretical optimal initialization methods, which are more efficient and stable than Gaussian or Uniform noise.

Publication	Data Initialization		Model Training		Grad Matching		Additional
	Distribution	Resolution	Network	Batch size	Loss-fn	Optimizer	
GradInv attacks of iteration-based framework							
DLG [Zhu <i>et al.</i> , 2019]	Gaussian	64×64	LeNet	8	ℓ_2 dist	L-BFGS	–
iDLG [Zhao <i>et al.</i> , 2020]	Uniform ^L	32×32	LeNet	1	ℓ_2 dist	L-BFGS	–
CPL [Wei <i>et al.</i> , 2020b]	Geometric	128×128	LeNet	8	ℓ_2 dist	L-BFGS	\mathcal{R}_y regularizer
InvGrad [Geiping <i>et al.</i> , 2020]	Gaussian ^L	224×224	ResNet ^T	8 (100)	Cosine	Adam	\mathcal{R}_{TV} regularizer
SAPAG [Wang <i>et al.</i> , 2020]	Constant	224×224	ResNet ^T	8	Gauss	AdamW	Gaussian kernel
GradInversion [Yin <i>et al.</i> , 2021]	Gaussian ^L	224×224	ResNet ^T	48	ℓ_2 dist	Adam	$\mathcal{R}_{\text{fidel}} + \mathcal{R}_{\text{group}}$
GradDisagg [Lam <i>et al.</i> , 2021]	Gaussian	32×32	MLP	32 (128)	ℓ_2 dist	L-BFGS	Participant info
GradAttack [Huang <i>et al.</i> , 2021]	Gaussian ^L	224×224	ResNet ^T	128	Cosine	Adam	No BN + labels
Bayesian [Balunović <i>et al.</i> , 2022]	Gaussian	32×32	ConvNet ^T	1 (32)	Cosine	Adam	Known $p(g x)$
CAFE [Jin <i>et al.</i> , 2021]	Uniform	32×32	Loop-Net	100	ℓ_2 dist	SGD	In Vertical-FL
GIAS [Jeon <i>et al.</i> , 2021]	Latent	64×64	ResNet ^T	4	Cosine	Adam	GAN-based
GradInv attacks of recursion-based framework							
PPDL-AHE [Phong <i>et al.</i> , 2018]	N/A	20×20	MLP	1	Gradient division		–
PPDL-SPN [Fan <i>et al.</i> , 2020]	N/A	32×32	ConvNet	8	Linear solving		Noise analysis
R-GAP [Zhu and Blaschko, 2021]	N/A	32×32	ConvNet	1	Inverse matrix		Rank analysis
COPA [Chen and Campbell, 2021]	N/A	32×32	ConvNet	1	Least-squares		Pull-back const

^L The labels can be directly identified or extracted from shared gradients.
^T The results of data recovery are compared in different model training states.

Table 1: Summary and classification of existing GradInv attacks.

Image Resolution. Without loss of generality, we consider the scenarios of image classification tasks. Actually, the resolution of raw images is an important factor that affects both initialization and the difficulty of recovery. The more pixels an image has, the more variables need to be optimized. Thus, it is more challenging to deploy attacks on complex datasets, such as ImageNet [Deng *et al.*, 2009]. In contrast, relatively good results can be achieved on the low-resolution black-and-white images (e.g., MNIST [LeCun, 1998], Fashion-MNIST [Xiao *et al.*, 2017]), even under mini-batch training. Till now, 224×224 pixels is the largest resolution for recovery.

Label Extraction. In a general process for iteration-based GradInv attacks, dummy data and labels are simultaneously updated. However, if the ground-truth labels can be extracted in advance, data recovery will be accelerated and the computational complexity will also be reduced. [Zhao *et al.*, 2020] first find out that the ground-truth label in a classification task can be directly revealed. For a mini-batch recovery, [Sun *et al.*, 2021; Yin *et al.*, 2021] propose that labels of different classes can be identified from each other, which assumes that there are non-repeating labels in the mini-batch training data. All of the above approaches extract the labels from the fully connected layer of gradients. Moreover, [Wainakh *et al.*, 2021a] exploit both the angle and magnitude of gradients to identify the labels. The angle shows whether a label is contained in the batch, while the magnitude indicates the number

of duplicate labels. [Dang *et al.*, 2021] present a more powerful label leakage attack, which can be applied to both image classification and speech recognition tasks. The labels can still be revealed after multiple local iterations.

Model Training for Gradient Generation

In order to obtain the generated gradients, the attacker needs to feed the initialized dummy data and labels into the model. Based on the error between the model outputs and the labels, the gradients of weights can be calculated through backpropagation. In the setting of distributed learning, the global model can be viewed as a white box, which means the model structure and weights are known. However, the depth of training network structures and the batch size of training data can implicitly affect the results of data recovery.

Network Model. Convolutional neural networks (CNN) or multilayer perceptron (MLP) are generally adopted as training networks for computer vision. Intuitively, the deeper the network is, the more parameters it contains. This raises two serious issues. First, the computational complexity is greatly increased, which makes it difficult or impossible for the optimization process to converge. Second, even if the procedure converges, there may exist multiple locally optimal solutions, resulting in a significant difference in the ground-truth value. So far, [Geiping *et al.*, 2020] are able to recover the raw data on ResNet-152 [He *et al.*, 2016], although only a few images can be recognized. [Yin *et al.*, 2021] achieve data recovery of

high-resolution images on ResNet-50, and display relatively better results. [Jin *et al.*, 2021; Balunović *et al.*, 2022] also investigate the relationship between the convergence states of model training and the errors of reconstruction.

Batch Training. For regular training, learning with mini-batch data can decrease the number of iterations and reduce the fluctuation of accumulated errors. However, this greatly increases the difficulty of GradInv attacks. Given an observation of gradients, the problem of recovering original data is equivalent to the decomposition of averaged summation. Using the vanilla optimization approach in Eq. (1), [Zhu *et al.*, 2019] can only perform data recovery for a maximum batch size of 8. With the assistance of various regularization terms, [Yin *et al.*, 2021; Huang *et al.*, 2021] all can recover the original private data with a batch size of over 30.

Data Update through Gradient Matching

The process of gradient matching measures the difference between generated gradients and ground-truth gradients, and then calculates the update of dummy inputs. Essentially, the procedure for data recovery can be analogized to supervised learning, which means the ground-truth gradients are similar to a high-dimensional “label”, while the dummy data and labels are the parameters to be learned in the optimization process. There are several important parts for gradient matching, the first is to obtain the gradients of a victim, and the next is proposing an effective method to minimize the distance.

Disaggregation. Considering the secure aggregation rules [Bonawitz *et al.*, 2017; Fereidooni *et al.*, 2021], the attacker can only observe a summation of the gradients. To perform gradient matching, it is necessary to decompose the individual gradients from the aggregation. Similar to the decomposition problem of mini-batch recovery, this is also a challenging problem. [Lam *et al.*, 2021] first focus on this issue and formulate it as a matrix factorization problem. By leveraging the participant information acquired from device analytics, additional constraints contribute to the solution. If G_{agg} represents the aggregated gradients, the factorization problem can be solved by finding the participant vector p_k :

$$\begin{aligned} \text{Find } p_k \text{ s.t. } & \text{Null}(G_{\text{agg}}^T)p_k = 0 \\ & C_k p_k - c_k = 0 \\ & p_k \in \{0, 1\}^n \end{aligned} \quad (3)$$

where $\text{Null}(\cdot)$ calculates the kernel of a matrix, and C_k specifies the participated rounds of client k .

Loss Function. Generally, an attacker is assumed to have direct access to the gradients of a victim. Thus, these studies focus on improving their algorithms to decrease the difference between gradients, and recover more realistic data. The enhancement consists of optimization metrics and regularization terms. To measure the distance between generated gradients and ground-truth gradients, Euclidean distance (i.e., ℓ_2 norm) is a frequently used loss function [Zhu *et al.*, 2019; Wei *et al.*, 2020b; Lam *et al.*, 2021]. However, [Wang *et al.*, 2020] discover that gradients with large values dominate data recovery at the early stages. They hence propose a weighted Gaussian kernel as the distance metric. Furthermore, [Geiping *et al.*, 2020] observe that the direction of gradients plays

a more important role than magnitude, and substitute the ℓ_2 cost function with cosine similarity:

$$\arg \min_{\mathbf{x}' \in [0, 1]^n} 1 - \frac{\langle \nabla W', \nabla W \rangle}{\|\nabla W'\| \|\nabla W\|} \quad (4)$$

Regularization. To recover more realistic data from the batch, some auxiliary regularization terms are inserted into the cost functions. These regularizers can be divided into two categories, one that constrains the fidelity of images, and the other that revises the position of the main object. Fidelity regularization [Geiping *et al.*, 2020; Yin *et al.*, 2021] steers the reconstructed data from impractical images, including total variation norm (TV), ℓ_2 norm and batch normalization (BN) [Nguyen *et al.*, 2015; Mahendran and Vedaldi, 2015]. Group consistency regularization is presented by [Yin *et al.*, 2021], which jointly considers multiple random seeds for initialization, and calculates the averaged data $\mathbb{E}(\mathbf{x}'_{g \in G})$ as reference. Any candidate whose recovery deviates from the “consensus” image of the group will be penalized.

$$\begin{aligned} \mathcal{R}_{\text{fidel}}(\mathbf{x}') &= \alpha_{\text{tv}} \mathcal{R}_{\text{TV}}(\mathbf{x}') + \alpha_{\ell_2} \mathcal{R}_{\ell_2}(\mathbf{x}') + \alpha_{\text{bn}} \mathcal{R}_{\text{BN}}(\mathbf{x}') \\ \mathcal{R}_{\text{group}}(\mathbf{x}', \mathbf{x}'_{g \in G}) &= \alpha_{\text{group}} \|\mathbf{x}' - \mathbb{E}(\mathbf{x}'_{g \in G})\|^2 \end{aligned} \quad (5)$$

Applying $\mathcal{R}_{\text{fidel}}$ and $\mathcal{R}_{\text{group}}$ regularizers can indeed support reconstructing more realistic image data on complex datasets (e.g., ImageNet) under mini-batch training.

2.2 Recursion-based Data Recovery

In recursion-based data recovery, the attacker can recursively calculate the input of each layer by finding the optimal solution with minimized error. [Phong *et al.*, 2018] first discover that the input of a perceptron can be directly recovered from: $x_k = \nabla W_k / \nabla b$. This conclusion is later generalized to fully connected (FC) layers or MLP, as long as bias terms exist. In accordance with this idea, [Fan *et al.*, 2020] convert a convolutional layer into a FC layer by stacking the filters, and then utilize the above relation. However, they neglect the feature of reused weights in the convolutional layers, which is totally different from that of FC layers. In order to recover the image data of the first convolutional layer, [Zhu and Blaschko, 2021] combine the forward and backward propagation, and formulate the problem as solving a system of linear equations. The essence of such data leakage is that: the feature map and the gradient of kernels in the first convolutional layer have direct involvement with the original data. Extending to the i th layer, it is possible to recover the input \mathbf{x}_i by solving:

$$\begin{cases} W_i \cdot \mathbf{x}_i = Z_i \\ \nabla Z_i \cdot \mathbf{x}_i = \nabla W_i \end{cases} \quad (6)$$

where $Z_i, \nabla Z_i$ represent the feature map and its gradients. Denoting the neuron outputs and activation function of each layer as a_i and $\sigma_i(\cdot)$, then we have the relationships of $Z_i = \sigma_i^{-1}(a_i)$ and $\nabla Z_i = \nabla a_i \cdot \sigma'_i(a_i)$. Hence, it is possible to recover the original data by recursively starting from the FC layer to the convolutional layer. [Chen and Campbell, 2021] also propose a generic framework by combining multiple optimization problems under different situations. The workflow of the recursion-based framework is depicted in Fig. 1(b)

Category	Method	Publication	Key Contribution
<i>Original Data</i>	MixUp	[Zhang <i>et al.</i> , 2018]	Data enhancement by linearly combining the inputs
	InstaHide	[Huang <i>et al.</i> , 2020]	Encrypt the MixUp data with one-time secret keys
	Pixelization	[Fan, 2018; Fan, 2019]	Perturb the raw data with pixelization-based method
<i>Training Model</i>	Dropout	[Zheng, 2021]	Add an additional dropout layer before the classifier
	Local iters	[Wei <i>et al.</i> , 2020b]	Share gradients after multiple local training iterations
	Architecture	[Zhu and Blaschko, 2021]	Reduce the number of convolutional kernels properly
<i>Shared Gradients</i>	Aggregation	[Zhang <i>et al.</i> , 2020]	Apply Homomorphic Encryption to protect gradients
		[Lia and Togan, 2020]	Utilize Secure Multi-Party Computation to aggregate
	Perturbation	[Sun <i>et al.</i> , 2021]	Perturb data representation layer and maintain utility
		[Wei <i>et al.</i> , 2021]	Add adaptive noise with differential privacy guarantee
	Compression	[Vogels <i>et al.</i> , 2019]	Compress the smaller values in gradients to zero
	[Karimireddy <i>et al.</i> , 2019]	Transmit the sign of gradients for model updates	

Table 2: Summary and classification of existing GradInv defense strategies.

Compared to the iteration-based approach, these recursion-based attacks have the following characteristics.

Non-initialization. Different from iterative optimization, a recursion-based attack can directly recover the original image without initialization or generating dummy inputs. Since the time of recovery is proportional to the square of the number of pixels, the current image resolution that can be recovered does not exceed 32×32 . Furthermore, the extraction of labels refers to the above-mentioned works in Section 2.1.

Model Structure. The networks for experiments include only convolutional and fully connected layers. The pooling layers or shortcut connections in ResNet are not considered, which would cause an accumulation of errors layer by layer, and make the reconstructed data extremely biased. However, these attacks can not deal with mini-batch training, which is a major difference compared to iteration-based attacks.

Linear Solving. As formulated in Eq. (6), the original data can be recursively solved by constructing linear equations. The feature maps and their gradients can be derived from the model weights and corresponding gradients. It is worth mentioning that such an attack depends on the integrity of gradients. Once the gradients are perturbed, the noisy solution will make the restored results completely unrecognizable.

3 Gradient Inversion Defenses

In this section, we summarize the emerging defense strategies from perspectives of *data obscuration*, *model improvement*, and *gradient protection*. A client can either obscure the private images from the data source, enhance the structure of network models, or protect the gradients before sharing. The main contributions of these defenses are described in Table 2.

3.1 Obscuration of Original Data

Since the target of GradInv attacks is to recover the training data of a victim, an ideal defense strategy is to directly protect

the raw data before training. We expect that the private input is difficult to be reconstructed while the model utility does not degrade too much. [Zhang *et al.*, 2018] propose the method of MixUp for data augmentation, where virtual training samples are generated by linearly combining a pair of data and labels. These generated examples can not only improve the accuracy of the training model, but also “aggregate” the original data. Based on MixUp, [Huang *et al.*, 2020] introduce the idea of cryptography to protect the data using one-time private keys. In particular, a portion of images from both private and public datasets are randomly selected for combination, and then the pixels are flipped according to the keys. This lightweight approach prevents an attacker from recovering the training data and ensures the usability of the data. [Pang *et al.*, 2020] also mixup the input with other clean samples to improve the adversarial robustness of training models. In addition, [Fan, 2019] protect the images with pixelization and Gaussian blur approaches, which can be used not only for distributed training, but also for data publication, such as crowd-sourcing.

3.2 Improvement of Training Model

As for training models, in addition to increasing the depth of the neural network or training with mini-batch (Section 2.1), we introduce some newly presented but effective approaches. In a general GradInv attack, it is assumed that the gradients are sent after one round of local training. [Wei *et al.*, 2020b] propose to schedule and control the number of locally training iterations before gradient sharing, which makes it more difficult to reconstruct the private data. Experiments demonstrate that the success rates of data recovery have dropped by more than 60% when performing 10 local iterations. Modifications to the network structure can also defend against the GradInv attacks. [Zheng, 2021] propose to add a simple dropout layer between the encoder and the classifier to solve the problem of overfitting. During the training process, there may be certain neurons with larger activation values, indicating the features

of training data are overly memorized. If a proportion of neurons are randomly pruned, then it is possible to mitigate the privacy inference attacks. Considering that different network models may have different risks of privacy leakage, [Zhu and Blaschko, 2021] present a rank-based security analysis. Such method indicates that the more filters in a convolutional layer, the better the data recovery will be. Similarly, [Geiping *et al.*, 2020] have also mentioned that it is impossible to recover the original data from gradients, if the dimensionality of model parameters is lower than that of input data. This conclusion inspires us to appropriately reduce the number of parameters while ensuring the performance of the model.

3.3 Protection from Gradient Sharing

In distributed learning, since model updates are performed on the basis of gradient exchanging, the straightforward privacy-preserving approach is to protect the gradients. Summarizing the existing studies, we divide them into: aggregation, perturbation and compression-based defense strategies.

Cryptography-based methods can generally guarantee the security and privacy of individual gradients without compromising their utility. [Bonawitz *et al.*, 2017; Lia and Togan, 2020] use secure multi-party computation (MPC) to compute the summation result of model updates. [Phong *et al.*, 2018; Kim *et al.*, 2018; Zhang *et al.*, 2020] implement Homomorphic Encryption (HE) to carry out operations on the ciphertext space for gradient aggregation. Even if an attacker steals the information through man-in-the-middle (MITM) attacks, he/she cannot decrypt it to obtain the ground-truth gradients. However, these approaches not only require modifications to the training architecture, but also exponentially increase the computation time, bandwidth and data storage.

Gradient perturbation is another frequently used approach for privacy protection. [Truex *et al.*, 2020; Wei *et al.*, 2020a; Yang, 2021] add Gaussian noise to the exchanged gradients with the guarantee of differential privacy (DP). [He *et al.*, 2021] essentially reveal how iterative training impacts privacy, and establishes the relationship between generalization and privacy-preserving. [Sun *et al.*, 2021] find the key to GradInv attacks lies in the data representation layer, and only perturb the gradient values in this layer. [Wei *et al.*, 2021] propose a method with dynamically adjustable noise which can achieve high resilience against GradInv attacks. Except for injecting noise, [Zhu *et al.*, 2019] discover that some compression methods, originally used to reduce communication overhead, can also be used to prevent data recovery. [Vogels *et al.*, 2019] propose to prune the smaller values to zero by a certain percentage, and [Karimireddy *et al.*, 2019] only transmit the sign of gradients. These methods can resist attacks to a certain extent while maintaining performance.

4 Conclusion and Future Directions

In this paper, we present recent advances in *Gradient Inversion* (GradInv) covering the offensive and defensive methods. To the best of our knowledge, for the first time, we review the GradInv attacks with a novel taxonomy and describe the main procedure for deploying an attack in distributed learning. We also reclassify the representative defense strategies,

which can be used to protect from data recovery. After summarizing the existing research on GradInv, we find that there are still some unresolved problems worthy of in-depth study. Finally, we discuss a few promising directions.

Combination of GradInv Attacks

We have introduced the characteristics of iteration-based and recursion-based attacks. The former can recover complex inputs from mini-batch data, while the latter enables one-shot and label-free reconstructions. [Qian *et al.*, 2021] present a two-step algorithm including both GradInv attacks, but only for a single convolutional layer without batch data. However, very few works have considered the combination of two kinds of attacks. If the constraint equations provided by recursion-based methods can be integrated into the iterative optimization procedure, we are able to improve the recovered results. Thus, it is necessary to propose a more powerful and efficient GradInv attack, which can speed up the recovery process or achieve a batch recovery on higher resolution datasets.

Extension to Transformers

Transformer [Vaswani *et al.*, 2017; Dosovitskiy *et al.*, 2021] is a recent popular network model that can be applied to both natural language processing and computer vision. Although GradInv attacks can be implemented on deep networks, such as ResNet, it is critical to extend from CNNs to Transformers. Since Transformer can handle more types of data, the attacks therein will cause more serious impacts. [Deng *et al.*, 2021] first attempts to deploy GradInv attacks on Transformer-based language models. [Lu *et al.*, 2021] propose to recover input of self-attention modules in Vision Transformer theoretically and empirically. Although each of these works has some limitations, it remains crucial and urgent to focus on the security and privacy issues in Transformers.

Quantification of Gradient Sharing

The defenses we have discussed are all active interventions to the training process. However, we can also indirectly analyze whether the gradients have a serious risk of privacy leakage. Since each user owns his/her training data and gradients, the extent of leakage can be quantified if any correlation between them is found. [Liu *et al.*, 2021] utilize mutual information to formulate this problem, and demonstrate the risk of gradient leakage is related to the model status and data distribution. [Mo *et al.*, 2021] propose \mathcal{V} -information and *sensitivity* metrics to quantify the layer-wise latent privacy leakages. This indicates that research in other domains can be transferred to study GradInv. We hope this survey can shed a light on future research and inspire more progress in different domains.

Acknowledgments

This research was supported by fundings from the Key-Area Research and Development Program of Guangdong Province (No. 2021B0101400003), Hong Kong RGC Research Impact Fund (No. R5060-19), General Research Fund (No. 152221/19E, 152203/20E, and 152244/21E), the National Natural Science Foundation of China (61872310), and Shenzhen Science and Technology Innovation Commission (JCYJ20200109142008673).

References

- [Balunović *et al.*, 2022] Mislav Balunović, Dimitar Iliev Dimitrov, Robin Staab, and Martin Vechev. Bayesian framework for gradient leakage. In *ICLR*, 2022.
- [Bonawitz *et al.*, 2017] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *CCS*, pages 1175–1191, 2017.
- [Bonawitz *et al.*, 2019] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. In *MLSys*, 2019.
- [Bouacida and Mohapatra, 2021] Nader Bouacida and Prasant Mohapatra. Vulnerabilities in federated learning. *IEEE Access*, 9:63229–63249, 2021.
- [Chen and Campbell, 2021] Cangxiong Chen and Neill D. F. Campbell. Understanding training-data leakage from gradients in neural networks for imageclassifications. In *NeurIPS Workshop*, 2021.
- [Dang *et al.*, 2021] Trung Dang, Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, Peter Chin, and Françoise Beaufays. Revealing and protecting labels in distributed training. In *NeurIPS*, 2021.
- [Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [Deng *et al.*, 2021] Jieren Deng, Yijue Wang, Ji Li, Chenghong Wang, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. Tag: Gradient attack on transformer-based language models. In *EMNLP*, 2021.
- [Dosovitskiy *et al.*, 2021] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- [Fan *et al.*, 2020] Lixin Fan, Kam Woh Ng, Ce Ju, Tianyu Zhang, Chang Liu, Chee Seng Chan, and Qiang Yang. Rethinking privacy preserving deep learning: How to evaluate and thwart privacy attacks. In *Federated Learning*, pages 32–50. Springer, 2020.
- [Fan, 2018] Liyue Fan. Image pixelization with differential privacy. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 148–162, 2018.
- [Fan, 2019] Liyue Fan. Differential privacy for image publication. In *Theory and Practice of Differential Privacy (TPDP) Workshop*, pages 1–6, 2019.
- [Fereidooni *et al.*, 2021] Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Helen Möllering, Thien Duc Nguyen, Phillip Rieger, Ahmad-Reza Sadeghi, Thomas Schneider, Hossein Yalame, et al. Safelearn: Secure aggregation for private federated learning. In *IEEE S&P Workshops (SPW)*, pages 56–62, 2021.
- [Geiping *et al.*, 2020] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients – how easy is it to break privacy in federated learning? In *NeurIPS*, pages 16937–16947, 2020.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [He *et al.*, 2021] Fengxiang He, Bohan Wang, and Dacheng Tao. Tighter generalization bounds for iterative differentially private learning algorithms. In *UAI*, 2021.
- [Huang *et al.*, 2020] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. In *ICML*, 2020.
- [Huang *et al.*, 2021] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. In *NeurIPS*, 2021.
- [Jegorova *et al.*, 2021] Marija Jegorova, Chaitanya Kaul, Charlie Mayor, Alison Q O’Neil, Alexander Weir, Roderick Murray-Smith, and Sotirios A Tsafaris. Survey: Leakage and privacy at inference time. *arXiv preprint arXiv:2107.01614*, 2021.
- [Jeon *et al.*, 2021] Jiwnoo Jeon, Kangwook Lee, Sewoong Oh, Jungseul Ok, et al. Gradient inversion with generative image prior. In *NeurIPS*, 2021.
- [Jin *et al.*, 2021] Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. Catastrophic data leakage in vertical federated learning. In *NeurIPS*, 2021.
- [Karimireddy *et al.*, 2019] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *ICML*, pages 3252–3261, 2019.
- [Kim *et al.*, 2018] Jinsu Kim, Dongyoung Koo, Yuna Kim, Hyunsoo Yoon, Junbum Shin, and Sungwook Kim. Efficient privacy-preserving matrix factorization for recommendation via fully homomorphic encryption. *ACM Transactions on Privacy and Security*, 21(4):1–30, 2018.
- [Lam *et al.*, 2021] Maximilian Lam, Gu-Yeon Wei, David Brooks, Vijay Janapa Reddi, and Michael Mitzenmacher. Gradient disaggregation: Breaking privacy in federated learning by reconstructing the user participant matrix. In *ICML*, 2021.
- [LeCun, 1998] Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998. Accessed: 2022-01-27.
- [Lia and Togan, 2020] Dragos Lia and Mihai Togan. Privacy-preserving machine learning using federated learning and secure aggregation. In *ECAI*, 2020.
- [Liu *et al.*, 2021] Yong Liu, Xinghua Zhu, Jianzong Wang, and Jing Xiao. A quantitative metric for privacy leakage in federated learning. In *ICASSP*, pages 3065–3069, 2021.
- [Lu *et al.*, 2021] Jiahao Lu, Xi Sheryl Zhang, Tianli Zhao, Xiangyu He, and Jian Cheng. April: Finding the achilles’ heel on privacy for vision transformers. *arXiv preprint arXiv:2112.14087*, 2021.

- [Mahendran and Vedaldi, 2015] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, pages 5188–5196, 2015.
- [McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agueria y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.
- [Mo *et al.*, 2021] Fan Mo, Anastasia Borovykh, Mohammad Malekzadeh, Hamed Haddadi, and Soteris Demetriou. Layer-wise characterization of latent information leakage in federated learning. In *ICLR Workshop*, 2021.
- [Nasr *et al.*, 2021] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papemoti, and Nicholas Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *IEEE Symposium on Security and Privacy (SP)*, pages 866–882, 2021.
- [Nguyen *et al.*, 2015] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, pages 427–436, 2015.
- [Pang *et al.*, 2020] Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *ICLR*, 2020.
- [Phong *et al.*, 2018] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2018.
- [Qian *et al.*, 2021] Jia Qian, Hiba Nassar, and Lars Kai Hansen. Minimal model structure analysis for input reconstruction in federated learning. In *NeurIPS Workshop*, 2021.
- [Sun *et al.*, 2021] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *CVPR*, 2021.
- [Truex *et al.*, 2020] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. Ldp-fed: Federated learning with local differential privacy. In *ACM International Workshop on Edge Systems, Analytics and Networking (EdgeSys)*, pages 61–66, 2020.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [Vogels *et al.*, 2019] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. Powersgd: Practical low-rank gradient compression for distributed optimization. In *NeurIPS*, pages 14259–14268, 2019.
- [Wainakh *et al.*, 2021a] Aidmar Wainakh, Fabrizio Ventola, Till Müßig, Jens Keim, Carlos Garcia Cordero, Ephraim Zimmer, Tim Grube, Kristian Kersting, and Max Mühlhäuser. User label leakage from gradients in federated learning. *arXiv preprint arXiv:2105.09369*, 2021.
- [Wainakh *et al.*, 2021b] Aidmar Wainakh, Ephraim Zimmer, Sandeep Subedi, Jens Keim, Tim Grube, Shankar Karuppayah, Alejandro Sanchez Guinea, and Max Mühlhäuser. Federated learning attacks revisited: A critical discussion of gaps, assumptions, and evaluation setups. *arXiv preprint arXiv:2111.03363*, 2021.
- [Wang *et al.*, 2020] Yijue Wang, Jieren Deng, Dan Guo, Chenghong Wang, Xianrui Meng, Hang Liu, Caiwen Ding, and Sanguthevar Rajasekaran. Sapag: A self-adaptive privacy attack from gradients. *arXiv preprint arXiv:2009.06228*, 2020.
- [Wei *et al.*, 2020a] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.
- [Wei *et al.*, 2020b] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating client privacy leakages in federated learning. In *ESORICS*, 2020.
- [Wei *et al.*, 2021] Wenqi Wei, Ling Liu, Yanzhao Wut, Gong Su, and Arun Iyengar. Gradient-leakage resilient federated learning. In *ICDCS*, pages 797–807, 2021.
- [Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [Yang, 2021] He Yang. H-fl: A hierarchical communication-efficient and privacy-protected architecture for federated learning. In *IJCAI*, 2021.
- [Yin *et al.*, 2021] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *CVPR*, pages 16337–16346, 2021.
- [Zhang *et al.*, 2018] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.
- [Zhang *et al.*, 2020] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *USENIX ATC*, pages 493–506, 2020.
- [Zhao *et al.*, 2020] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.
- [Zheng, 2021] Yanchong Zheng. Dropout against deep leakage from gradients. *arXiv preprint arXiv:2108.11106*, 2021.
- [Zhu and Blaschko, 2021] Junyi Zhu and Matthew B. Blaschko. R-gap: Recursive gradient attack on privacy. In *ICLR*, 2021.
- [Zhu *et al.*, 2019] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *NeurIPS*, 2019.