Zhichao You, Xuewen Dong, Shujun Li, Ximeng Liu, Siqi Ma and Yulong Shen, "Local Differential Privacy is Not Enough: A Sample Reconstruction Attack against Federated Learning with Local Differential Privacy," IEEE Transactions on Information Forensics & Security, Volume 20, pp. 1519-1534, IEEE, 2024
doi: 10.1109/TIFS.2024.3515793

Local Differential Privacy is Not Enough: A Sample Reconstruction Attack against Federated Learning with Local Differential Privacy

Zhichao You, Xuewen Dong, *Member, IEEE,* Shujun Li, *Senior Member, IEEE,*
Ximeng Liu, *Senior Member, IEEE,* Siqi Ma, *Member, IEEE,* Yulong Shen, *Member, IEEE*

***Abstract*—Reconstruction attacks against federated learning (FL) aim to reconstruct users' samples through users' uploaded gradients. Local differential privacy (LDP) is regarded as an effective defense against various attacks, including sample reconstruction in FL, where gradients are clipped and perturbed. Existing attacks are ineffective in FL with LDP since clipped and perturbed gradients obliterate most sample information for reconstruction. Besides, existing attacks embed additional sample information into gradients to improve the attack effect and cause gradient expansion, leading to a more severe gradient clipping in FL with LDP. In this paper, we propose a sample reconstruction attack against LDP-based FL with any target models to reconstruct victims' sensitive samples to illustrate that FL with LDP is not flawless. Considering gradient expansion in reconstruction attacks and noise in LDP, the core of the proposed attack is gradient compression and reconstructed sample denoising. For gradient compression, an inference structure based on sample characteristics is presented to reduce redundant gradients against LDP. For reconstructed sample denoising, we artificially introduce zero gradients to observe noise distribution and scale confidence interval to filter the noise. Theoretical proof guarantees the effectiveness of the proposed attack. Evaluations show that the proposed attack is the only attack that reconstructs victims' training samples in LDP-based FL and has little impact on the target model's accuracy. We conclude that LDP-based FL needs further improvements to defend against sample reconstruction attacks effectively.**

I. Introduction

FEDERATED learning (FL) is a distributed learning framework in which users train a given global model through local samples without uploading these samples to the server or the platform [1]–[3]. The server updates the global model according to gradients or model updates uploaded by users. Since the server trains machine learning models without collecting users' training samples, FL effectively protects users' privacy and reduces communication overhead. FL has been applied to multiple platforms for user privacy preservation, e.g., word prediction of Google Gboard [4], automatic speech recognition of Apple's Siri [5], and credit evaluation of WeBank [6].

(a) (b) (c) (d) (e)

Fig. 1. Samples reconstructed by different attacks when the batch size is 16: (a) reconstructed by an existing attack in FL without LDP; (b) reconstructed by the same attack in FL with LDP; (c) reconstructed by the proposed attack in FL with LDP; (d) a linear combination of generated samples; (e) generated samples with noise in the background.

Reconstruction attacks against gradients uploaded by users cause a severe training sample leakage issue in FL (e.g., [7]–[12]). Although users do not share private training samples, the information about the samples is implied in the gradients generated by the local samples. Combined with reconstruction attacks, adversaries can reconstruct users' original samples according to the uploaded gradients with high accuracy and quality, which is shown in Fig. 1(a).

Existing reconstruction attacks are lacking in feasibility and practicability. On the one hand, these attacks make too strong assumptions for better reconstruction attack performance (e.g., [13]–[15]). For example, a common assumption is that the adversary has multiple statistics of victims' training samples (e.g., [14], [16], [17]). However, victims have no motivation to calculate and upload the sample features, let alone how the adversary obtains the sample features from victims. On the other hand, existing works are difficult to extend to practical situations. Many attacks are only effective in simple cases with small batch sizes or training samples with low pixels (e.g., [7], [8], [12], [18]). In contrast, the batch size and pixels are both larger in practice. Moreover, some attacks restrict the setting of the target model (e.g., [18]–[20]), damaging the target model's performance and making these attacks impracticable.

Besides, FL with local differential privacy (LDP) can protect users' privacy against reconstruction attacks (e.g., [21]–[24]), in which gradients are clipped and perturbed. As shown in Fig. 1(b), existing attacks hardly work in FL mechanisms with LDP. These attacks' ineffectiveness comes from two

This work was supported in part by the National Key R&D Program of China (No. 2023YFB3107500), National Natural Science Foundation of China (No. 62220106004, 62232013), Technology Innovation Leading Program of Shaanxi (No. 2022KXJ-093, 2023KXJ-033), and Innovation Fund of Xidian (No. YJSJ24015). (Corresponding author: Xuewen Dong.)

Zhichao You, Xuewen Dong and Yulong Shen are with the School of Computer Science & Technology, Xidian University, and are with the Shaanxi Key Laboratory of Network and System Security, Xi'an, China (email: zcyou@stu.xidian.edu.cn, xwdong@xidian.edu.cn, ylshen@mail.xidian.edu.cn).

Shujun Li is with the School of Computing and the Institute of Cyber Security for Society (iCSS), University of Kent, Canterbury, UK (e-mail: S.J.Li@kent.ac.uk).

Ximeng Liu is with the College of Computer and Data Science, Fuzhou University, Fuzhou, China (e-mail: snbnix@gmail.com).

Siqi Ma is with the School of System and Computing, University of New South Wales, Canberra, ACT 2612, Australia (e-mail: siqi.ma@unsw.edu.au).

TABLE I
SETTING OF THE PRIVACY PARAMETER ε AND DEVIATIONS OF DIFFERENT METRICS IN EXISTING FL MECHANISMS WITH LDP.

Reference	Wei's work [21]			Zhou's work [22]		
Metric	Loss function			Model accuracy		
ε	6	8	10	1	5	10
Deviations	120%	85%	57%	52%	27%	24%

LDP measures: clipping and perturbing gradients. Some attacks modify target models so that the gradient contains a large amount of sample information, resulting in a *gradient expansion*, i.e., the norm of gradients is too large. When the gradients are clipped and perturbed, the sample information in the gradients is seriously damaged, leading to the failure of the reconstruction attack. Existing attacks rely on accurate gradient values to reconstruct training samples. When a small amount of noise is added to these gradients, the reconstructed samples differ from the actual ones. We make a more detailed theoretical analysis of the failure of the existing attacks in Section V-A.

This paper aims to determine **whether an adversary can feasibly and practically reconstruct the samples of victims in FL with LDP**. Table I provides a comparison of the privacy parameter ε settings in existing FL mechanisms with LDP, highlighting the deviations caused by noise to the loss function or model accuracy in the worst case, compared to scenarios without DP. The results indicate that when $\varepsilon = 10$, the noise introduced by LDP significantly reduces the model accuracy.[1] For smaller ε, the model performance drops so dramatically that the global model becomes unusable. However, our simulation results demonstrate that a malicious server can still effectively reconstruct the most private samples of victims in FL with LDP through the proposed attack (as shown in Fig. 1(c)) when $\varepsilon = 10$.

Theoretical analysis ensures the effectiveness of the proposed attack. Experimental results show that the proposed attack can effectively reconstruct sensitive information of samples from clipped and perturbed gradients protected by LDP. As shown in Fig. 1(c), the reconstructed sample of the proposed attack against protected gradient exposes the primary information of the sample. In contrast, other attacks can only reconstruct meaningless noise. In addition, the proposed attack has almost no impact on the model training of non-target users, which ensures the performance of the target model. The key contributions of this paper are as follows:

- **Reconstruction attack against FL with LDP.** The proposed attack is the first reconstruction attack that reconstructs user samples in FL mechanisms with LDP, where user gradients are clipped and perturbed. Additionally, the attack targets only specific victims, having minimal impact on non-target users' training and the performance of global models.
- **Attack feasibility and practicability.** The adversary does not interfere with the FL training process but targets standard FL mechanisms without requiring additional abilities or knowledge. Furthermore, the proposed attack is flexible, remaining effective across different target models, training samples, and large batch sizes. Consequently, this attack can be applied to any FL protocol and various learning scenarios while maintaining satisfactory concealment.
- **Techniques against LDP.** We propose several techniques to counter noise introduced by LDP for sample reconstruction in FL with LDP. Firstly, we introduce a separation layer to prevent gradient expansion caused by reconstruction attacks, reducing the information loss of clipping gradients in LDP. This approach retains minimal gradient information necessary for sample reconstruction and employs an image segmentation model to extract the main subjects from samples, significantly reducing the gradients' norm. Secondly, we enhance the quality of the reconstructed samples by incorporating an imprinted structure that observes the noise distribution and scales the confidence interval to mitigate background noise. Additionally, we introduce a metric saver that imprints sample metrics onto the gradients, which are then used as an optimization objective to improve sample quality.
- **Theoretical and experimental validity proof.** We guarantee the effectiveness of the proposed attack through theoretical analysis. Evaluation results demonstrate that the attack can effectively reconstruct users' privacy information of training samples in FL with LDP. Evaluations indicate that the proposed attack has minimal impact on FL training and target model accuracy. Additionally, we analyze several factors influencing the attack's performance, including privacy parameters and model complexity. Through evaluation results, we identify the critical conditions under which the attack is most effective in FL with LDP. Finally, we discuss the limitations of the proposed attack and suggest possible defenses.

[1]Results come from experiments of the corresponding papers.

II. RELATED WORK

Sample reconstruction attacks are mainly divided into the following categories: optimization-based attacks (e.g., [7], [8], [17], [25]), network-based attacks (e.g., [16], [18], [26], [27]), and analysis-based attacks (e.g., [10], [11], [28]). Table II provides a brief comparison of these efforts.

Optimization-based Attacks. In optimization-based attacks, the adversary regards reconstructed samples as multiple random variables and optimizes the reconstructed samples through gradients of objective functions. DLG [7] first proposed an optimization-based sample reconstruction attack, in which the objective function is the ℓ^2 norm of the difference between victims' gradients and reconstructed samples-generated gradients. Other existing works proposed various improvements according to different requirements. For example, Yin et al. [9] considered ℓ^2 norm of the gradient difference, total variation of reconstructed samples, and sample statistics difference in the objective function to reconstruct samples with higher quality. Since the reconstructed samples are considered as multiple variables, the effectiveness of the

TABLE II
COMPARISON OF EXISTING SAMPLE RECONSTRUCTION ATTACKS IN FL.

Attack Type	Mechanism	Effectiveness in FL with LDP	Sample Resolution	Notes
Optimization-based Attack	DLG [7]	×	64×64	The first effective reconstruction attack through gradients.
	Yin's work [9]	×	224×224	Considering multiple metrics for optimization.
	Pan's work [25]	×	224×224	Discussion about model complexity and attack effectiveness.
Network-based Attack	mGAN-AI [18]	×	64×64	Samples must be identically distributed.
	GIAS [16]	×	64×64	Applying the GAN to generate image prior.
Analysis-based Attack	Fowl's work [10]	×	224×224	Constructing bins to separate training samples.
	Boenisch's work [11]	×	224×224	Considering passive and active attacks in different scenarios.
Hybrid Attack	The Proposed Attack	√	224×224	The sole effective reconstruction attack in FL with LDP.

optimization-based attacks depends heavily on the complexity of the sample (i.e., the number of random variables) and the complexity of the model (i.e., the number of constraints). When the batch size is large, finding the optimal solution (completely reconstructed samples) in optimization-based attacks is difficult.

Network-based Attacks. The adversary in network-based reconstruction attacks trains a deep learning model to reconstruct victims' training samples. Most network-based attacks apply generative adversarial networks (GAN) [29] to generate images similar to users' original samples. An attack mGAN-AI [18] modifies the global model with GAN in FL so that victims train the generator in the global model while training the classifier with local samples. The adversary reconstructs victims' training samples through the converged generator. GIAS [16] applied GAN to transform the sample variables in the optimization problem into the input variables of the generator with lower dimensions, significantly reducing the search space and facilitating the finding of a better solution. Khosravy et al. [26] used a similar idea to successfully reconstruct victims' facial features in a face recognition system. A significant challenge for network-based attacks is to obtain enough samples with the same distribution as users' samples to train the generator.

Analysis-based Attacks. In analysis-based attacks, the adversary reconstructs training samples through the exact connection between gradients and training samples. Based on the theorem that gradients can calculate the input of FCL, Fowl et al. [10] proposed to add a linear FCL to the global model and separate sample gradients so that most of the victims' samples can be reconstructed with high quality. Besides, based on the connection between gradients and training samples, Franziska et al. [11] proposed a passive attack for reconstructing a single sample and an active attack for reconstructing multiple samples, where another idea for modifying the global model is given. The analysis-based attacks reconstruct high-quality training samples with low complexity and are still effective in the case of large batch sizes.

Some reconstruction attacks modify the global model structure to enable gradients to contain more sample information and improve the quality of reconstructed samples (e.g., [10], [11], [18], [26], [27]). Figure 2 compares sample reconstruction attacks using original and modified models. By embedding malicious structures in the global model, the gradients generated by modified models and client data carry additional sensitive information. Since the adversary reconstructs victims' samples through their gradient, attacks with modified models can achieve better attack effects than attacks with original models.

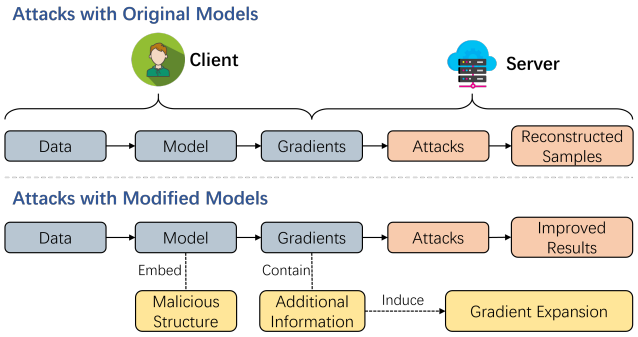


Fig. 2. Comparison of sample reconstruction attacks using original and modified models.

A disadvantage of attacks with modified models is causing gradient expansion, i.e., the norm of gradients increases since gradients carry additional information. Section V-A provides a technical analysis for generating gradient expansion. Gradient expansion presents challenges when reconstructing samples in FL with LDP. Since gradients are clipped according to a fixed norm value and perturbed with noise, gradients with additional information are severely compressed. As a result, the signal-to-noise ratio in the protected gradient drops significantly. No existing works can effectively reconstruct samples from clipped and perturbed gradients. The proposed attack against FL with LDP modifies the global model to obtain effective performance while reducing the impact of gradient expansion by gradient compression presented in Sections V-A and V-B.

In addition, existing attacks rely on accurate gradients to reconstruct samples. In FL with LDP, the noise in the perturbed gradient causes existing attacks to reconstruct meaningless noise samples. The proposed attack reduces the impact of perturbed gradients on the attack effect by filtering the noise of gradients and reconstructed samples presented in Sections V-C

and V-D. Dealing with gradient clipping and perturbation in LDP by gradient compression and noise filtering, the proposed attack effectively reconstructs sensitive information in the sample from the protected gradients in FL with LDP.

III. Preliminaries

A. Federated Learning

FL is a distributed learning framework to solve the privacy concern that servers (i.e., model owners) train their models with users' sensitive and private data. In FL, servers design machine Learning target models based on predicted task requirements. Then, they distribute the global model (i.e., a target model) to users who train the global model with their local data to produce intermediate training results (gradients or new parameters). Finally, the server aggregates the intermediate training results and obtains new global model parameters. The above procedures are repeated until global models converge.

We illustrate FL framework details with a typical FL algorithm FedSGD [30], the FL aggregation algorithm considered in this paper. Specifically, a server customizes the global ML model structure f, initialized model parameters ω, hyper-parameters, and training process. The server distributes the training materials to users whose local data is x and y. User i trains the model with a random batch of its local data x_i and y_i and generates model gradients $\nabla_\omega L(f(x_i;\omega), y_i) = \partial L(f(x_i;\omega), y_i)/\partial\omega$, where L is a loss function. The server aggregates users' uploaded gradients and updates the global model parameters distributed to users for the next round of training. The training process terminates until the global model converges or meets the termination condition.

B. Federated Learning with Local Differential Privacy

Existing works apply LDP [31] to protect users' intermediate results during the FL training process (e.g., [21]–[24], [32]). LDP enables users to perturb gradients with noise before uploading gradients to the server.

Definition 1 (Differential privacy [31])**.** *A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ)-differentially private if for all $S \subseteq \mathrm{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:*

$$\Pr[\mathcal{M}(x) \in S] \leq \exp(\varepsilon) \Pr[\mathcal{M}(y) \in S] + \delta. \quad (1)$$

Adjusting privacy parameters (ε, δ) can meet various user privacy and data accuracy requirements. In most cases, better user privacy protection would reduce data accuracy.

As in the existing works [21]–[23], [33], we consider that users add the Gaussian noise $\mathcal{N}(0, \sigma^2)$ to gradients for privacy preservation. Given privacy parameters (ε, δ), the Gaussian mechanism is (ε, δ)-differentially private when setting a proper scale σ [31]. Algorithm 1 presents a general local learning process for users in FL mechanisms with LDP. Users train the global model with a batch of local samples and generate local gradients. Then, generated gradients are clipped according to the clipping bound to avoid the norm of generated gradients being too large so that the perturbation is too small to protect user privacy. Users add the Gaussian noise to clipped gradients with the scale factor decided by themselves according to different privacy requirements and upload the perturbed gradients to the server.

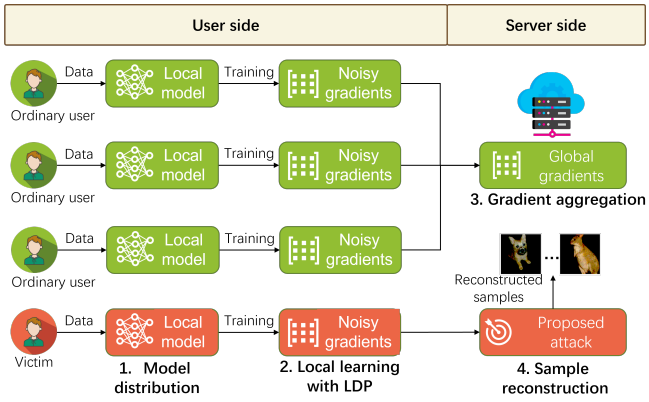


Fig. 3. The process of the proposed attacks in FL with LDP.

Algorithm 1: Local training with LDP

Input: Global model f, model parameters ω, loss function L, a batch of samples $\{x, y\}$, clipping bound C, and scale factor σ

Output: Clipped and noisy local gradient $\nabla_\omega L$

Generate local gradients $\nabla_\omega L = \frac{\partial L(f(x;\omega),y)}{\partial\omega}$;

Clip gradients $\nabla_\omega L = \nabla_\omega L / \max\left(1, \frac{\|\nabla_\omega L\|}{C}\right)$;

Perturb gradients $\nabla_\omega L = \nabla_\omega + \mathcal{N}(0, \sigma^2)$;

return $\nabla_\omega L$

C. Sample Reconstruction Attacks in Federated Learning

Figure 3 introduces the process by which the server implements the proposed reconstruction attacks and steals user samples through clipped and perturbed gradients uploaded by users with LDP. Firstly, the server distributes the global model structure and parameters to users as local models for local training in model distribution, which is a standard setting of FL. Then, users train local models with their data and generate local gradients. In FL with LDP, users clip and add noise to the generated local gradients according to custom privacy parameters before uploading the gradients to the server for gradient protection. Therefore, the server only gets clipped and perturbed gradients.

For ordinary users, i.e., non-target users, the server aggregates their gradients to update the global model to ensure model performance. For the victim, i.e., the target user, the server reconstructs its samples according to its clipped and perturbed gradients through the proposed attack (given in Section V and Section VI). It should be noted that existing reconstruction attacks cannot reconstruct the victim's samples through clipped and perturbed gradients. Experimental results in Section VII show that even though protecting gradients by LDP, the proposed attack is still effective in reconstructing the victim's samples.

Users upload local gradients instead of original samples to avoid privacy leaks in FL. However, sample reconstruction attacks can reconstruct users' training samples through their uploaded gradients, exposing privacy risks in FL. Meanwhile, the proposed attack shows that FL with LDP cannot entirely defend sample reconstruction attacks.

IV. Threat Model and Primary Attack

A. Threat Model

Attack motivations. The privacy protection FL provides users is that users can perform model training by leaving sensitive samples locally. However, training samples contain precious information, such as medical, financial, or personal information. When model training and performance are not affected, the server may be interested in performing an attack to secretly reconstruct training data to obtain more benefits.

Adversary's goal. The adversary performs a reconstruction attack to reconstruct users' training samples from gradients and extract the sensitive information in the samples. For example, as shown in Fig. 1(c), we assume that the adversary is more concerned with the subject information in the image, and its background information can be ignored. However, the adversary cannot require the user to pre-process local samples before training to reduce the difficulty of the attack. Meanwhile, whether a reconstruction attack is performed should not be distinguished from the performance of the global model on target prediction tasks for better attack concealment.

Adversary's ability. The malicious server's only ability is to design a global model for training, which is a common situation in FL (e.g., [1], [24], [30]). Besides, designing a model is also the basic assumption of most existing reconstruction attacks (e.g., [10], [11], [34]). We do not limit target models and prediction tasks for better attack practicability, i.e., the designed global model should completely contain any given target models. This paper aims to show that LDP-based FL without global model verification cannot protect users' privacy.

Adversary's knowledge. The adversary has the complete structure and parameters since they are in charge of global design. In addition, the adversary receives victims' clipped and noisy gradients with LDP protection, which is discussed as follows. Meanwhile, the server only obtains victims' gradients once, i.e., the attack should be effective with one round of gradients, and the adversary cannot require victims to upload multiple rounds of gradients to reduce attack difficulty.

Gradient protection. We considers gradient protection with LDP based on the setting in Section III-B, which contains clipping and perturbation of the gradient as shown in Algorithm 1. Users determine the privacy parameters and clipping bounds in LDP, and the above values are not exposed to the server.

B. Primary Attack

The primary attack considers a simple case on a fully connected layer (FCL) where the batch size is one. More complicated cases with larger batch sizes and complex models are discussed in Section V.

Lemma 1. *The adversary can reconstruct the input of any FCL through its gradients when the batch size is one [35].*

Fig. 4. Reconstructed training samples generated by the primary attack when the batch size is 2, 4, 8, and 16.

Proof. Suppose the parameters of an FCL are $[w, b]^\intercal$, for any single sample x, the output of the FCL is $y = w^\intercal x + b$. According to the chain rule, we have

$$\nabla_b L = \frac{\partial L}{\partial b} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial b} = \frac{\partial L}{\partial y}, \tag{2}$$

and

$$\nabla_w L = \frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial w} = \frac{\partial L}{\partial y} \cdot x = \nabla_b L \cdot x. \tag{3}$$

As a result, the adversary can reconstruct the sample x by $x = \nabla_w L \oslash \nabla_b L$ where $\oslash$ is the entry-wise division. □

Lemma 1 provides a straightforward way of reconstructing samples from gradients: embedding an FCL to the model so that the first layer is an FCL and reconstructing samples by Lemma 1, which we refer to as the *primary reconstruction attack*. However, the batch size is hardly set to one in a real scenario, and the primary reconstruction attack is limited when the batch size is larger due to the following theorem.

Theorem 1. *The output of the primary attack is a linear combination of training samples.*

Proof. Suppose that the batch size is B, and training samples are $\left\{x^{(1)}, x^{(2)}, \cdots, x^{(B)}\right\}$, according to the back-propagation, we have

$$\nabla_b L = \frac{1}{B}\sum_{i=1}^{B} \frac{\partial L}{\partial b} = \frac{1}{B}\sum_{i=1}^{B} \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial b} = \frac{\partial L}{\partial y}, \tag{4}$$

and

$$\nabla_w L = \frac{1}{B}\sum_{i=1}^{B} \frac{\partial L}{\partial w} = \frac{1}{B}\sum_{i=1}^{B} \frac{\partial L}{\partial y} \cdot x^{(i)} = \frac{\nabla_b L}{B}\sum_{i=1}^{B} x^{(i)}. \tag{5}$$

As a result,

$$\nabla_w L \oslash \nabla_b L = \frac{1}{B}\sum_{i=1}^{B} x^{(i)}, \tag{6}$$

i.e., a linear combination of all training samples. □

Theorem 1 shows that when the batch size exceeds 1, the primary attack only gets a linear combination of training samples. When the batch size is large, such a linear combination cannot expose too much information about the training samples. For example, Fig. 4 shows the training samples reconstructed by the primary attack when the batch size is 2, 4, 8, and 16. As the batch size increases, it is harder to distinguish sample information from the reconstructed samples.

A straightforward solution is to let each neural unit in the FCL only contain one sample's gradients, and the adversary can reconstruct separated training samples through the primary

attack. Specifically, suppose that there are K units in the FCL, and unit k only contains the gradients generated by sample $x^{(i)}$, we have

$$\nabla_{b_k} L = \frac{1}{B} \cdot \frac{\partial L}{\partial b_k} = \frac{1}{B} \cdot \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial b_k} = \frac{1}{B} \cdot \frac{\partial L}{\partial y_k}, \quad (7)$$

and

$$\nabla_{w_k} L = \frac{1}{B} \cdot \frac{\partial L}{\partial w_k} = \frac{1}{B} \cdot \frac{\partial L}{\partial y_k} \cdot x^{(i)} = \nabla_{b_k} L \cdot x^{(i)}. \quad (8)$$

When unit k only contains the gradients generated by $x^{(i)}$, the adversary can reconstruct $x^{(i)}$ by $\nabla_{w_k} L \oslash \nabla_{b_k} L$ as discussed in Lemma 1, where $\oslash$ is the entry-wise division. For convenience, we refer to $\nabla_w L$ and $\nabla_b L$ as weight and bias gradients, respectively. According to the primary attack, the adversary can reconstruct any sample with its weight and bias gradients. We refer to the above process of separating gradients of each sample into different units of FCL as *gradient separation*.

Since gradients are clipped and perturbed, the primary attack cannot obtain effective reconstructed results in FL with LDP. The primary attack cannot distinguish the samples' corresponding gradients (weight and bias gradients) and noise from FCL gradients. It further leads to the failure of gradient separation. Besides, $x^{(i)}$ is reconstructed by perturbed $\nabla_w L$ and $\nabla_b L$, which leads to the primary attack getting some meaningless noise samples since gradients are noisy.

V. Reconstruction Attack against FL with LDP

A. Gradient Separation without Expansion

We first briefly analyze the reason for gradient expansion. As discussed in Section IV-B, the primary attack reconstructs sample $x^{(i)}$ by $\nabla_{w_k} L \oslash \nabla_{b_k} L$. According to Theorem 1, the key of the primary attack is to make each neural unit in the FCL only contain one sample's gradient, i.e., gradient separation. Therefore, a larger number of units in the FCL brings better effectiveness of gradient separation. For example, the existing attack [10] requires about 1024 units to separate gradients of samples in ImageNet dataset [36] when the batch size is 16. Since the sample size is $16 \times 3 \times 224 \times 224$, the existing attacks introduce additional $1024 \times 16 \times 3 \times 224 \times 224$ values into the gradients, resulting in a large norm of gradients and causing gradient expansion.

We propose a gradient separation method against FL with LDP, which has the following advantages. Increasing the number of units in FCLs can improve the separation effect of the proposed method but would not increase the norm of gradients, effectively avoiding the gradient expansion. The following are the technical implementation details.

Suppose an FCL contains K units with parameters $[w, b]^\intercal$, the weights of all units in the FCL are set to equal, i.e., $w_1 = w_2 = \cdots = w_K$ (its value is a hyper-parameter given in Section VII). The bias parameters in the FCL are set according to the quantile function of a random variable following the Laplace distribution. In other words, given $X \sim \text{Laplace}(\mu, s)$, $b_j = -F_X^{-1}(j/K)$ for bias of unit j, where $s > 0$ is a scale factor and $F_X(\cdot)$ is the cumulative distribution function (CDF) of X. The output of the FCL y_{min} is the minimum positive value of $(w^\intercal x + b)$. We refer to an FCL with the above setting as a *separation layer* and introduce its variations to achieve the proposed attack in Section VI.

The separation layer achieves the following property for gradient separation without gradient expansion. A *reverse index* $i_0 \in \{1, 2, \ldots, K\}$ of any sample $x^{(i)}$ is an index of the unit in the separation layer such that $w_{i_0}^\intercal x^{(i)} + b_{i_0} > 0$ and $w_{i_0+1}^\intercal x^{(i)} + b_{i_0+1} \leq 0$. The i_0-th unit in the separation layer is a *reverse unit* of sample $x^{(i)}$.

Theorem 2. *Each sample's corresponding gradients in the separation layer only exist in its reverse units.*

Proof. Bias of unit j in the FCL is set to $b_j = -F_X^{-1}(j/K)$ where $X \sim \text{Laplace}(0, s)$ and $F_X(\cdot)$ is the CDF of X, and we have $b_1 > b_2 > \cdots > b_K$. Since $w_1 = w_2 = \cdots = w_K$, for any sample $x^{(i)}$,

$$w_1^\intercal x^{(i)} + b_1 > w_2^\intercal x^{(i)} + b_2 > \cdots w_K^\intercal x^{(i)} + b_K. \quad (9)$$

Combining the reverse index i_0, we have

$$\begin{cases} w_1^\intercal x^{(i)} + b_1 > w_2^\intercal x^{(i)} + b_2 > \cdots > w_{i_0}^\intercal x^{(i)} + b_{i_0} > 0; \\ \\ w_K^\intercal x^{(i)} + b_K < \cdots < w_{i_0+1}^\intercal x^{(i)} + b_{i_0+1} \leq 0. \end{cases} \quad (10)$$

Suppose $y_{\text{min}}^{(i)}$ is the minimal non-zero positive value of $w^\intercal x^{(i)} + b$, $y_{\text{min}}^{(i)} = w_{i_0}^\intercal x^{(i)} + b_{i_0}$. In other words, for any sample $x^{(i)}$, $y_{\text{min}}^{(i)}$ depends on the i_0-th unit in the FCL, i.e., $w_{i_0}^\intercal x + b_{i_0}$. When only considering sample $x^{(i)}$, we have

$$\begin{cases} \nabla_{w_{i_0}} L = \frac{\partial L}{\partial w_{i_0}} = \frac{\partial L}{\partial y_{\text{min}}^{(i)}} \cdot \frac{\partial y_{\text{min}}^{(i)}}{\partial w_{i_0}} = \frac{\partial L}{\partial y_{\text{min}}^{(i)}} \cdot x^{(i)}; \\ \\ \nabla_{b_{i_0}} L = \frac{\partial L}{\partial b_{i_0}} = \frac{\partial L}{\partial y_{\text{min}}^{(i)}} \cdot \frac{\partial y_{\text{min}}^{(i)}}{\partial b_{i_0}} = \frac{\partial L}{\partial y_{\text{min}}^{(i)}}. \end{cases} \quad (11)$$

On the other hand, for any unit $k' \neq i_0$ in the FCL,

$$\begin{cases} \nabla_{w_{k'}} L = \frac{\partial L}{\partial w_{k'}} = \frac{\partial L}{\partial y_{\text{min}}^{(i)}} \cdot \frac{\partial y_{\text{min}}^{(i)}}{\partial w_{k'}} = 0; \\ \\ \nabla_{b_{k'}} L = \frac{\partial L}{\partial b_{k'}} = \frac{\partial L}{\partial y_{\text{min}}^{(i)}} \cdot \frac{\partial y_{\text{min}}^{(i)}}{\partial b_{k'}} = 0. \end{cases} \quad (12)$$

The above equations lead to Theorem 2. □

More intuitively, Theorem 2 proves that sample $x^{(i)}$ only generates gradients at the i_0-th unit in the separation layer, which achieves gradient separation. The adversary can reconstruct samples from gradients of reverse units in the separation layer when reverse units only contain one sample's corresponding gradients. Specifically, for any sample $x^{(i)}$, suppose that unit i_0 only contains the gradients of sample $x^{(i)}$, the adversary can reconstruct $x^{(i)}$ by $\nabla_{w_{i_0}} L \oslash \nabla_{b_{i_0}} L$.

Meanwhile, according to the following theorem, the number of units with non-zero gradients in the separation layer is not more than the batch size, preventing gradient expansion. Taking an example on ImageNet dataset [36] where the batch size is 16 and samples are images with $3 \times 224 \times 224$ pixels, the number of non-zero gradients in the separation layer is not greater than $16 \times 3 \times 224 \times 224$. However, the number of non-zero gradients in the existing method [10] is $1024 \times 3 \times 224 \times 224$.

Theorem 3. *Even multiple increases in the number of units in the separation layer reduce the probability that each unit contains multiple sample gradients, and the number of units with non-zero gradients is not greater than the batch size in the separation layer.*

Proof. We first analyze the probability that the reverse index of any sample $x^{(i)}$ is k, i.e., $\Pr\{i_0 = k\}$. Recall that the bias parameters of the FCL in the inference structure are set by the quantile function of a variable following the Laplace distribution, we have $b_j = -F_X^{-1}(j/K)$ where $X \sim \text{Laplace}(0, s)$, $F_X(\cdot)$ is the CDF of X, and $n = K$ is the number of units in the FCL. Specifically, according to the CDF of Laplace distribution,

$$F_X^{-1}\left(\frac{j}{K}\right) = -s \cdot \text{sgn}\left(\frac{j}{K} - 0.5\right) \ln\left(1 - 2\left|\frac{j}{K} - 0.5\right|\right). \tag{13}$$

For any sample $x^{(i)}$,

$$\begin{aligned} y_k &= w_k^\intercal x^{(i)} + b_k = w_k^\intercal x^{(i)} - F_X^{-1}\left(\frac{k}{K}\right) \\ &= w_k^\intercal x^{(i)} + s \cdot \text{sgn}\left(\frac{k}{K} - 0.5\right) \ln\left(1 - 2\left|\frac{k}{K} - 0.5\right|\right). \end{aligned} \tag{14}$$

Without loss of generality, assume that $(k/K - 0.5) > 0$,

$$y_k = w_k^\intercal x^{(i)} + s \cdot \ln\left(2 - \frac{2k}{K}\right). \tag{15}$$

Therefore, we have

$$\begin{aligned} &\Pr\{i_0 = k, n = K\} = \Pr\{y_k > 0 \text{ and } y_{k+1} \le 0\} \\ =&\Pr\left\{p\left(\frac{2k}{K}\right) < w_k^\intercal x^{(i)} \le p\left(\frac{2(k+1)}{K}\right)\right\} \\ =&\Pr\left\{p\left(\frac{2\cdot 2k}{2K}\right) < w_k^\intercal x^{(i)} \le p\left(\frac{2(2k+1)}{2K}\right)\right\} + \\ &\Pr\left\{p\left(\frac{2(2k+1)}{2K}\right) < w_k^\intercal x^{(i)} \le p\left(\frac{2\cdot 2(k+1)}{2K}\right)\right\} \\ =&\Pr\{i_0 = 2k, n = 2K\} + \Pr\{i_0 = 2k+1, n = 2K\}, \end{aligned} \tag{16}$$

since we set that $w_1 = w_2 = \cdots w_n$, abbreviating $-s \cdot \ln(2 - 2k/K)$ to $p(2k/K)$.

The above results show that $i_0 = k$ means that $x^{(i)}$ belongs to $[p(2k/K) \oslash w_k^\intercal, p(2(k+1)/K) \oslash w_k^\intercal]$, which we refer to *reverse interval*. Increasing n to $2K$ is equivalent to dividing the reverse interval into two parts.

Given two samples $x^{(i)}$ and $x^{(j)}$, we have $i_0 \neq j_0$ when $n = 2K$ if $i_0 \neq j_0$ when $n = K$ since the reverse intervals of i_0 and j_0 are not disjoint. Otherwise, assume that $i_0 = j_0 = k$ when $n = K$, increasing n to $2K$ makes i_0 and j_0 change to $2k$ or $2k+1$, resulting in four combinations of i_0 and j_0. However, $i_0 = j_0$ only occurs in two of these combinations. Increasing n from K to $2K$ reduces the probability that i_0 equals j_0. When gradients of all training samples are separated into different units, the reverse indexes of the samples are $\{1_0, 2_0, \cdots, B_0\}$ where B is the batch size. In other words, only units in the above reverse index set have non-zero gradients, and $|\{1_0, 2_0, \cdots, B_0\}| = B$. □

Theorem 3 shows that the adversary can increase the number of units to separate gradients as much as possible, and the growth of units would not cause gradient expansion.

B. Removing Background Gradients

We further compress gradients by removing background gradients to reduce the compressed degree of gradients in clipping. As shown in Fig. 1(c), the image's subject is a dog, while the background is worthless to the adversary. Inspired by this, we propose keeping the pixels where subjects are located before samples enter the separation layer while the rest are set to 0. The above operation makes the gradient corresponding to the pixel with a value of 0 in the image also be 0 in the separation layer.

The implementation of subject extraction is based on the segment anything model (SAM) [37]. SAM has significant advantages in image segmentation, and many machine learning models apply SAM to improve target models' performance [38]–[40]. Most importantly, SAM is a zero-shot model, i.e., SAM can directly apply to all user samples without users performing any training on SAM.

SAM generates masks for any image to segment the image with multiple input modes. We set the center of images as the selecting (input) points and apply masks with higher scores to samples. Pixels in the mask with the highest score are kept, while the rest will be set to 0 for gradient compression.

C. Sample Denoising

Since samples only retain subjects through masks generated by SAM while other pixels (i.e., backgrounds) are set to 0, the background pixels in reconstructed samples should also be 0. The sample denoising aims to restore the background pixels of reconstructed samples to 0 through noise filtering.

We first analyze the cause of noise in the backgrounds of reconstructed samples. Consider a pixel $p_{c,w,h}^{(i)}$ in the background of sample $x^{(i)}$, after subject extraction, $p_{c,w,h}^{(i)}$ is set to 0. Assuming that the relevant gradient is scaled by ω in gradient clipping, the reconstructed value $\hat{p}_{c,w,h}^{(i)}$ can be calculated as follows when the perturbation is not considered:

$$\hat{p}_{c,w,h}^{(i)} = \frac{\nabla_{w_{i_0,c,w,h}} L/\omega}{\nabla_{b_{i_0}} L/\omega} = \frac{\nabla_{w_{i_0,c,w,h}} L}{\nabla_{b_{i_0}} L}, \tag{17}$$

where i_0 is the reverse unit, and $w_{i_0,c,w,h}$ and b_{i_0} are the corresponding weight and bias, respectively. When $p_{c,w,h}^{(i)} = 0$, we have $\nabla_{w_{i_0,c,w,h}} L = 0$ and $\hat{p}_{c,w,h}^{(i)} = 0$. When gradients are perturbed,

$$\hat{p}_{c,w,h}^{(i)} = \frac{0 + n_w}{\nabla_{b_{i_0}} L/\omega + n_b} \neq 0, \tag{18}$$

where n_w and n_b are noise for gradient perturbation, and this is why there is noise in the background of reconstructed samples.

Gradient filtering introduces extra zero gradients that reflect the noise distribution after perturbation to obtain the noise's confidence interval. As shown in Fig. 5, before passing through the separation layer, samples pass through a convolution layer with a kernel size of 1, a step size of 1, and an output channel of 6 without bias term. The weights of the first 3 channels

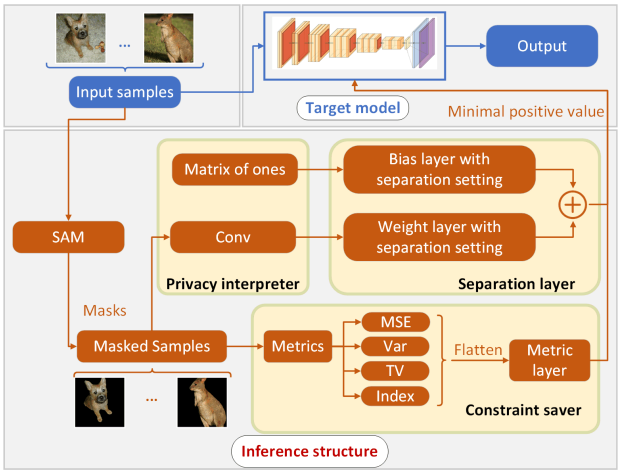


Fig. 5. The framework of the global model with any target models in the proposed attack.

and the last 3 channels are set to 1 and 0, respectively. The output of this layer is the same sample as processed samples and a vector of all zeros with the same size as the processed samples. The adversary can collect noise samples and get the noise distribution after perturbation from the corresponding gradients of introduced zero gradients.

Specifically, since the noise in LDP follows a normal distribution, the noise with negative values satisfies a half-normal distribution with a mean of $-\sigma\sqrt{2}/\sqrt{\pi}$ and a variance of $\sigma^2\left(1-2/\pi\right)$ where σ is the standard deviation of the noise [41]. Extra zero gradients are artificially added to gradients, and there are enough noise samples in the gradient to infer σ by the mean and variance of noise according to the law of large numbers (LLN) [42]. Further, we can generate a confidence interval $[-c, c]$ for noise by σ.

Note that the noise in backgrounds is scaled with different factors according to Equation (18), i.e., n_w is scaled by $\left(\nabla_{b_{i_0}} L/\omega + n_b\right)$, meaning the confidence interval also should be scaled. We propose an improved structure and modify the bias term of the primary inference structure so that bias gradients are repeated and identical. Specifically, as shown in Fig. 5, we delete the bias term of the FCL in the separation layer (i.e., the *weight layer*) and use a *bias layer* without the bias term to generate bias of the weight layer. The weights of the bias layer are set to the quantile function of a random variable following the Laplace distribution, which is mentioned in Section V-A, and its input is a matrix of ones. The weight and bias layer output are added to the target model's input. The improved separation layer comprising the weight and bias layers satisfies the following theorem.

Theorem 4. *In the improved separation layer, for any sample $x^{(i)}$, weight gradients of sample $x^{(i)}$ only exists in i_0-th unit of the weight layer, and gradients of the bias layer are repeated and identical bias gradients of sample $x^{(i)}$.*

Proof. The weights of the weight layer are $w_1 = w_2 = \cdots = w_K$ where K is the number of units. Suppose that the input size of the bias layer is D, the weights of the bias layer are set according to $b_{j,k} = -F_X^{-1}(k/K)$ where $X \sim \text{Laplace}(0, s)$, $F_X(\cdot)$ is the CDF of X and $j \in [1, D]$. The weight layer is an FCL without bias term, given any input $x^{(i)}$, the output of the weight layer is $\left\{w_1^\intercal x^{(i)}, w_2^\intercal x^{(i)}, \ldots, w_K^\intercal x^{(i)}\right\}$. Since the bias layer is an FCL without bias term, the input of the bias layer is a vector of ones, and $b_{1,k} = b_{2,k} = \cdots = b_{D,k}$, the output of the bias layer is

$$\sum_{j=1}^{D} \{b_{j,1}, b_{j,2}, \ldots, b_{j,K}\} = D \cdot \{b_{1,1}, b_{1,2}, \cdots, b_{1,K}\}. \quad (19)$$

We have $b_{1,1} > b_{1,2} > \cdots > b_{1,K}$ according to the proof of Theorem 2, and $D \cdot b_{1,1} > D \cdot b_{1,2} > \cdots > D \cdot b_{1,K}$. Thus, any sample $x^{(i)}$ has a unique reverse index i_0 such that

$$\begin{cases} w_1^\intercal x^{(i)} + D \cdot b_{1,1} > \cdots > w_{i_0}^\intercal x^{(i)} + D \cdot b_{1,i_0} > 0; \\ w_K^\intercal x^{(i)} + D \cdot b_{1,K} < \cdots < w_{i_0+1}^\intercal x^{(i)} + D \cdot b_{1,i_0+1} \leq 0. \end{cases} \quad (20)$$

The output of the improved inference structure only retains the minimal non-zero positive value, i.e., $y_{\min}^{(i)} = w_{i_0}^\intercal x^{(i)} + \sum_{j=1}^{D} b_{j,i_0}$. Similar to the proof of Theorem 3, weight gradients of sample $x^{(i)}$ only exist in the i_0-th unit of the weight layer which is the only unit in the weight layer that affects the value of $y_{\min}^{(i)}$. For bias gradients, we have

$$\begin{cases} \nabla_{w_{i_0}} L = \frac{\partial L}{\partial w_{i_0}} = \frac{\partial L}{\partial y_{\min}^{(i)}} \cdot \frac{\partial y_{\min}^{(i)}}{\partial w_{i_0}} = \frac{\partial L}{\partial y_{\min}^{(i)}} \cdot x^{(i)}; \\ \nabla_{b_{j,i_0}} L = \frac{\partial L}{\partial b_{j,i_0}} = \frac{\partial L}{\partial y_{\min}^{(i)}} \cdot \frac{\partial y_{\min}^{(i)}}{\partial b_{j,i_0}} = \frac{\partial L}{\partial y_{\min}^{(i)}}. \end{cases} \quad (21)$$

Any gradients of b_{j,i_0} where $j \in [1, D]$ are bias gradients of sample $x^{(i)}$, and we have

$$\nabla_{b_{1,i_0}} L = \cdots = \nabla_{b_{D,i_0}} L = \frac{\partial L}{\partial y_{\min}^{(i)}}. \quad (22)$$

In other words, there are D identical bias gradients in the improved inference structure. The adversary can reconstruct sample $x^{(i)}$ by

$$x^{(i)} = \nabla_{w_{i_0}} L \oslash \left(\frac{1}{D}\sum_{j=1}^{D} \nabla_{b_{j,i_0}} L\right). \quad (23)$$

□

The intuition of the proof is that the bias of the weight layer changes from one value to multiple identical values (the number is equal to the input size of the bias layer). Therefore, we can obtain an accurate $\nabla_{b_{i_0}} L/\omega$ by averaging the corresponding gradients in the bias layer. For example, assume that the weight layer has 1024 units and the input size of the bias layer is 500, $\nabla_{b_{i_0}} L/\omega + n_b$ appears once in the primary inference structure. However, in the improved separation layer, $\left(\nabla_{b_{i_0}} L/\omega + n_b\right)$ appears 500 times in the gradients. Although noise in the repeated bias gradients (n_b) is different, the averaging effectively realizes noise cancellation according to the LLN because their mean is 0.

Finally, for any sample $x^{(i)}$, we scale the confidence interval $[-c, c]$ by the averaged $\nabla_{b_{i_0}} L/\omega$. When the value of a specific pixel $\hat{p}_{c,w,h}^{(i)}$ is in the interval, we think that $\hat{p}_{c,w,h}^{(i)}$ has a

high probability of being 0, and set $\hat{p}_{c,w,h}^{(i)}$ to 0, which can effectively filter the noise in backgrounds.

D. Metric-based Optimization

The above noise filtering is mainly aimed at noise in backgrounds, and we further propose metric-based optimization to improve the quality of reconstructed samples. We set the optimization objective as

$$\min_{\hat{x}} w_\mu \sum_l \|\mu_l(\hat{x}) - \mu_l(x)\|_2 + w_\sigma \|\sigma_l^2(\hat{x}) - \sigma_l^2(x)\|_2 + w_{\mathrm{TV}} \sum_l \|\mathrm{TV}_l(\hat{x}) - \mathrm{TV}_l(x)\|_2, \tag{24}$$

where $\mu_l(\cdot)$, $\sigma_l^2(\cdot)$, and $\mathrm{TV}_l(\cdot)$ are sample-wise mean, variance and total variation, respectively, and w_μ, w_σ, and w_{TV} are weight coefficients. The problem is obtaining the above information of processed samples for optimization.

As shown in Fig. 5, we introduce a *metric layer* to imprint the above information to gradients. Specifically, the model calculates the corresponding batch-wise metrics of processed samples to generate a metric matrix, which then is flattened to become the metric layer input. The output of the metric layer is directly added to the input of the target model. As discussed in Lemma 1, the adversary can reconstruct the input of any FCL through its gradients when the batch size is 1 (the flattened metric matrix can be viewed as a single sample). For example, a batch of 16 images with 3 channels can generate a $16 \times 3 \times 3$ metric matrix, which is then flattened to 1×144 as an input of the metric layer.

Another problem is that reconstructed samples are ordered by samples' reverse units, but metrics are ordered by input samples, which makes the order of reconstructed samples and metrics inconsistent. The metrics are sample-wise, and the non-corresponding order would optimize reconstructed samples in the wrong direction. Therefore, we also save sample reverse units in the gradients of the metric layer. The corresponding gradients can reflect the reverse units of samples, and the adversary can reorder the metrics according to these gradients so that the orders of metrics and reconstructed samples are consistent. In addition, the metric layer introduces a repeated gradient structure as the bias layer to realize noise cancellation and improve accuracy. Finally, the adversary can optimize the reconstructed samples according to ordered metrics according to Equation (24).

VI. All-in-one and Implementation

A. Model Setting and Distribution

We introduce the global model with any target models in which the inference structure is embedded through Fig. 5. In the local training process, samples pass through the SAM and the target model. The gradients in the target model are not affected by the proposed attack for non-target users, ensuring the accuracy of the converged target model. As discussed in Section V-B, the SAM generates masked samples to remove background gradients. The masked samples are then sent to both the convolution and metric layers. The output of the convolution layer is the input of the weight layer, and the input of the bias layer is a matrix of ones. As discussed in Section V, the above setting leaves sample information in the gradient without gradient expansion so the server can load the sample information from gradients for reconstruction attack. The gradients of the metric layer are used to optimize the reconstructed samples, as discussed in Section V-D. When the user is a victim, the input of the target model is a linear combination of the outputs of weight, bias, and metric layers. For non-target users, the input of the target model is the original samples, which means that the inference structure does not impact the training of the target model.

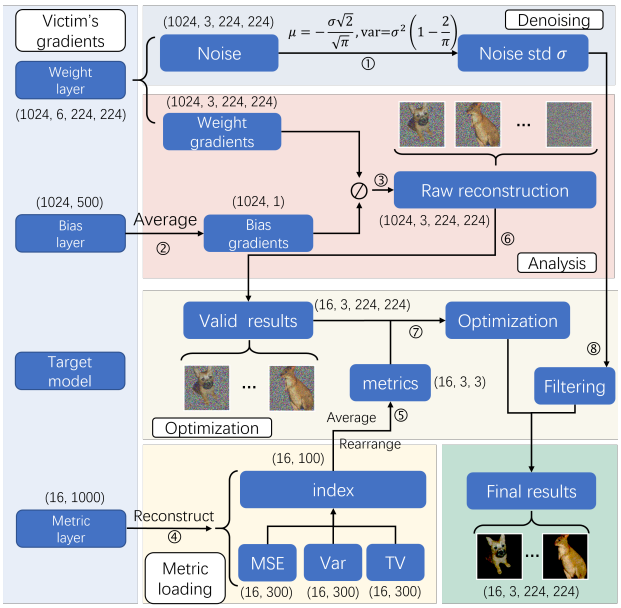


Fig. 6. The process of reconstructing training samples according to the victim's gradients with an example where the batch size is 16 and the sample size is $3 \times 224 \times 224$. The number of units in the weight, bias, and metric layers are 1024, 500, and 1000, respectively.

In the model distribution phase, the adversary distributes the global model with the target model and the designed inference structure, as shown in Fig. 5, to non-target users and the victim. In order to prevent target model training of non-target users from being affected by the attack, the output coefficient of the inference structure of non-target users is set to small. The inference structure has little effect on non-target users, and the final model output of non-target users is almost consistent with the target model. The victim's gradient would be ignored in the gradient aggregation phase. Considering the thousands of users in FL, ignoring the victim's gradients has almost no effect on the aggregated gradient. We discuss the impact of the proposed attack on FL training and target model accuracy in Section VII-E through simulations. The victim generates gradients by training the global model with local samples, then clips and perturbs the gradients according to local privacy parameters and uploads the processed gradients to the server.

B. Implementation

Figure 6 shows the process of implementing the proposed sample reconstruction attack through the clipped and perturbed

gradients uploaded by the victim. We cover each implementation detail step by step in the following.

(1) Privacy parameters extraction. As discussed in Section V-C, the convolutional layer adds extra zero gradients the same size as training samples to the gradients of the weight layer. Through the mean and variance of negative gradients of the structure that introduces extra zero gradients before perturbation, the adversary obtains the standard variance σ of the noise in gradient perturbation according to the mean of $-\sigma\sqrt{2}/\sqrt{\pi}$ and the variance of $\sigma^2\left(1-2/\pi\right)$.

(2) Bias term reconstruction. As discussed in Section V-C, the gradients of the bias layer are used to generate accurate bias gradients. Since bias gradients are repeated and identical in the gradients of the bias layer, we can obtain more accurate bias gradients by averaging.

(3) Raw reconstruction. In addition to noisy zero gradients, there are noisy weight gradients in the gradients of the weight layer. According to an element-wise division of weight and bias gradients discussed in Section IV-B, i.e., $\nabla_w L \oslash \nabla_b L$, the adversary performs a raw reconstruction for training samples.

(4) Metric reconstruction. As discussed in Section V-D, the metric layer gradients contain sample metrics, including mean, variance, total variation, and samples' reverse units. The adversary reconstructs the above metrics and samples' reverse units from metric layer gradients.

(5) Metric alignment. The order of reconstructed metrics and reconstruction samples is different, and the order of reconstructed samples can be inferred from the index of reverse units. Therefore, reconstructed metrics are reordered according to the reconstructed index of reverse units so that the orders of metrics and reconstructed samples are consistent.

(6) Image filtering. Since the number of units with non-zero gradients is not greater than the batch size according to Theorem 2, raw reconstruction contains many meaningless images. For example, there are 1008 meaningless images in the raw reconstruction in Fig. 6. However, these meaningless images are easy to distinguish, and reverse units reconstructed from the gradients of the metric layer provide accurate positions of valid samples in the raw reconstruction.

(7) Metric-based optimization. As discussed in Section V-D, the proposed attack establishes the optimization objective by reconstructed metrics and optimizes the reconstructed samples. The proposed attack optimizes the valid samples after image filtering according to Equation (24) to improve the reconstructed sample quality.

(8) Noise filtering. Generating confidence interval for noise according to the reconstructed standard variance σ and scaling the confidence interval through average bias gradients of reverse units to filter the reconstructed samples' noise.

VII. Evaluation

A. Evaluation Setup

Benchmark. We utilize the benchmark on Breaching, an open framework for reconstruction attacks against FL[2]. We consider two analysis-based attacks (Fowl's attack [10] and Boenisch's attack [11]) in the evaluation. Both can almost completely reconstruct victims' training samples in FL mechanisms without LDP. We also introduce two optimization-based attacks for comparison (Yin's attack [9] and Wei's attack [12]). Breaching implements the above attacks. In addition, Hong's work [43] discusses model vulnerability through the Hessian matrix to gradient difference. We implement the optimization algorithm in Hong's work [43] for comparison.

Datasets and models. We consider four image datasets: ImageNet [36], CIFAR-100 [44], Caltech-256 [45], and Flowers102 [46]. Users' training samples are randomly selected from the above datasets, and training samples are normalized between 0 and 1. The default target model is ResNet101 [47].

Metrics. Following metrics measure the quality of reconstructed samples: mean square error (MSE), peak signal-to-noise ratio (PSNR), and complex wavelet structural similarity (CW-SSIM) [48]. MSE reflects the difference between two images. PSNR quantifies reconstruction quality for images subject to lossy compression, and a higher PSNR generally indicates that given images have a higher reconstruction quality. CW-SSIM is an index that varies between 0 and 1 to measure the similarity of two images, and a larger CW-SSIM refers to a higher similarity between two images. The above metric calculates the similarity between masked training samples and reconstructed samples generated by the proposed attack [49], [50].

Parameter setting. The Gaussian noise scale is calculated based on the setting of the existing FL mechanism with LDP [21], [22], [33], i.e., $\sigma_U = 2cC/m\varepsilon$ where c is a constant, C is the clipping bound, m is the minimal size of local datasets, and ε is the privacy parameter. The default values of the above parameters are close to those in the experiments of existing works, specifically, $c = 1$, $C = 10$, $m = 1000$, $\varepsilon = 10$, and $\delta = 0.01$. As shown in Table I, setting ε to 10 in the existing FL mechanisms with LDP causes significant performance degradation to the target model. Smaller ε will make the target model wholly inapplicable due to low accuracy. The data presented in the evaluation is the average of 10 results under the same conditions. The adversary implements attacks through a round of gradients. Other parameters are given in Table VIII of Appendix A.

Experiment environment. All experiments are conducted on a system equipped with an NVIDIA RTX 4090D GPU with 24GB memory and an Intel Xeon Platinum 8474C CPU. The proposed attack is implemented by PyTorch, and all computations are accelerated by CUDA.

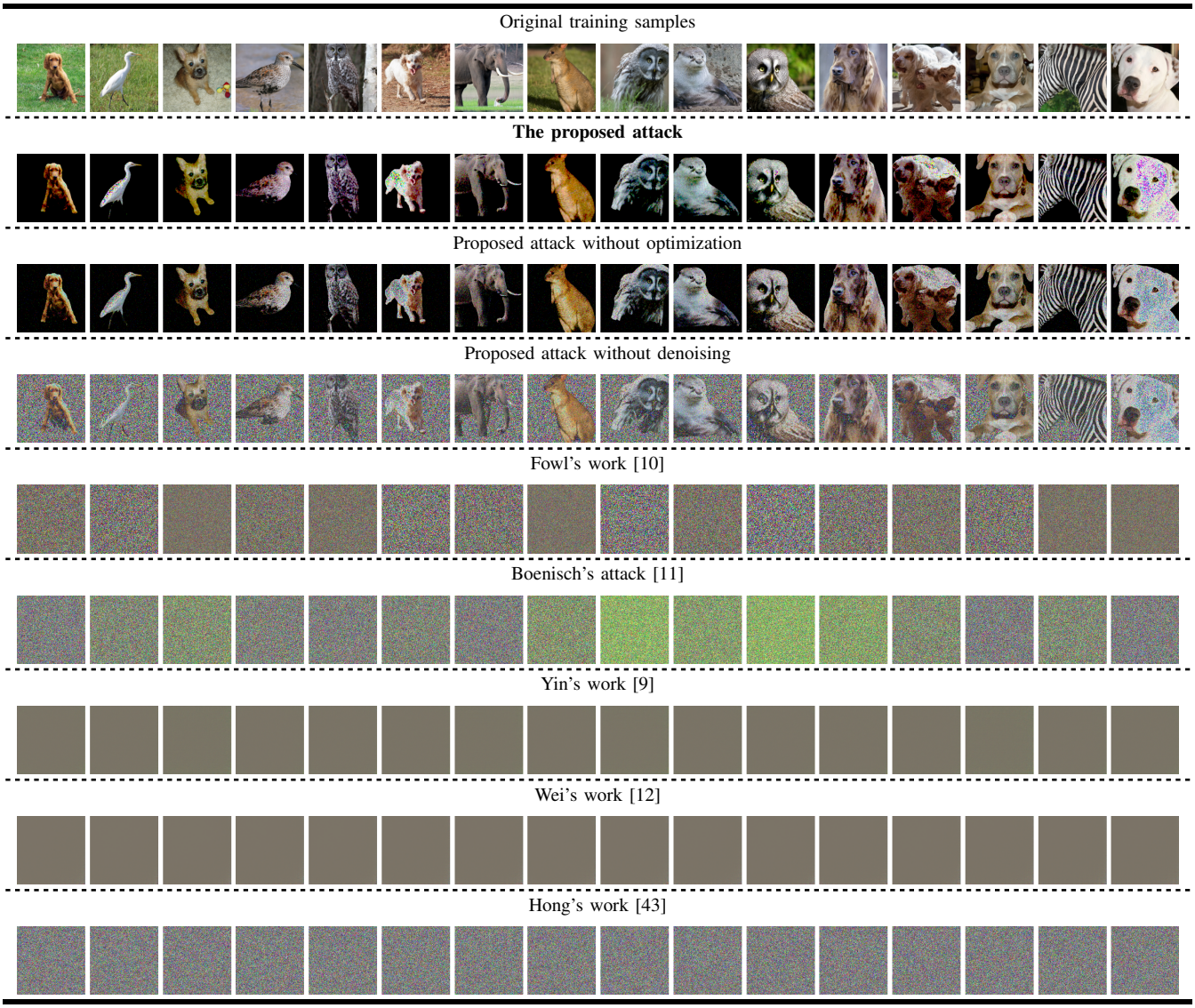
B. Reconstructed Samples Quality

We compare the reconstructed samples' quality of various attacks when gradients are clipped and perturbed in FL with LDP. The victim has the same samples when the adversary performs different attacks to compare the effectiveness of different attacks. We first set the batch size to 16 and discuss the impact of batch sizes on the performance in Section VII-C. The global target model is ResNet101 [47], while both the clipping bound and ε are set to 10.

Table III compares the victim's original samples from ImageNet and reconstructed samples generated by various attacks.

[2]https://github.com/JonasGeiping/breaching.

TABLE III
ORIGINAL TRAINING SAMPLES FROM IMAGENET AND RECONSTRUCTED SAMPLES GENERATED BY DIFFERENT ATTACKS IN FL WITH LDP.



Data in random reconstruction is randomly generated by a uniform distribution from 0 to 1. Most samples reconstructed by the proposed attack can provide the primary information in the training samples. Meanwhile, there is a slight noise in the reconstructed samples, and some reconstructed images cannot present any information since LDP protects victims' gradients. Other sample reconstruction attacks hardly reconstruct the victim's samples when gradients are clipped and perturbed.

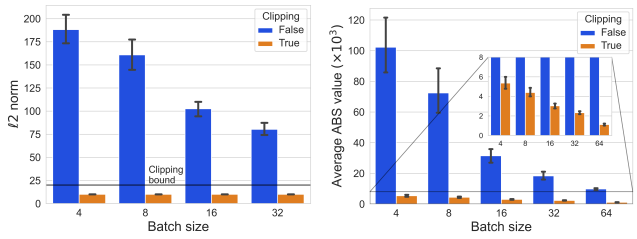
Table IV compares the quality of samples reconstructed by various attacks under different datasets. *Without optimization* is the result of the proposed attack without metric-based optimization. *Without denoising* is the result of the proposed attack without image denoising (noise filtering and metric-based optimization). The above two settings are to present the effect of noise filtering and metric-based optimization on the proposed attack. Furthermore, Table X in Appendix A compares the quality of reconstructed samples of different attacks against gradients without LDP protection, clipped gradients, and perturbed gradients, respectively. Figure 7 provides a more straightforward presentation of training samples reconstructed by the proposed attack when different LDP materials protect victims' gradients. Most attacks can extract victims' sensitive information from reconstructed samples without LDP protection. The analysis-based attacks (the proposed attack, Fowl's attack [10], and Boenisch's attack [11]) almost wholly reconstruct the original samples.

(a) (b) (c) (d) (e) (f)

Fig. 7. Reconstructed samples generated by the proposed attack under different gradient protections: (a) the original sample; (b) masked samples; (c) gradients protected by clipping and perturbation; (d) no gradient protection; (e) gradients protected by clipping; (f) gradients protected by perturbation.

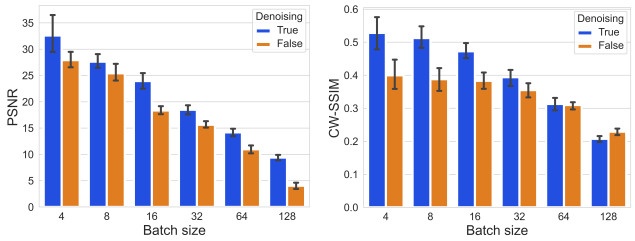
TABLE IV
QUALITY OF SAMPLES RECONSTRUCTED BY VARIOUS ATTACKS UNDER DIFFERENT DATASETS IN FL WITH LDP.

Attack	ImageNet			CIFAR100			Caltech256			Flowers102		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Proposed attack	**0.0002**	**28.520**	**0.836**	**0.0018**	**26.869**	**0.868**	**0.0016**	**29.670**	**0.853**	**0.0017**	**29.705**	**0.881**
Without optimization	0.0004	25.800	0.481	0.0039	25.799	0.482	0.0039	25.806	0.482	0.0039	25.778	0.481
Without denoising	0.0067	21.805	0.384	0.0069	21.795	0.384	0.0068	21.803	0.385	0.0070	21.785	0.384
Fowl's attack [10]	0.319	4.973	0.204	0.316	5.015	0.199	0.318	4.994	0.203	0.332	4.794	0.191
Boenisch's attack [11]	0.168	8.234	0.213	0.164	8.531	0.134	0.199	7.503	0.185	0.186	7.760	0.195
Yin's attack [9]	0.067	12.129	0.139	0.069	12.443	0.153	0.099	10.677	0.113	0.130	8.947	0.196
Wei's attack [12]	0.063	12.179	0.093	0.115	11.036	0.127	0.099	10.652	0.108	0.127	9.924	0.109
Hong's work [43]	0.3140	6.249	0.069	0.436	6.377	0.053	0.327	4.528	0.061	0.216	4.320	0.094
Random guess	0.155	8.165	0.214	0.156	8.213	0.231	0.176	7.665	0.187	0.174	7.659	0.186



(a) $\ell 2$ norm of gradients (b) Absolute value of gradients

Fig. 8. The $\ell 2$ norms and absolute values of gradients under different conditions.



(a) PSNR (b) CW-SSIM

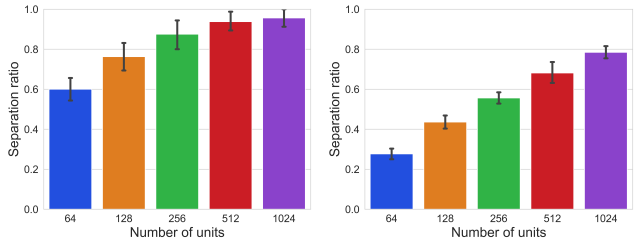
Fig. 9. Impact of batch size on reconstructed image quality.

We also find that protecting gradients by clipping or perturbation alone has little effect on the proposed attack. Gradients are clipped by $\nabla_\omega L = \nabla_\omega L / \max\left(1, \frac{\|\nabla_\omega L\|}{C}\right)$, and samples are reconstructed by $x = \nabla_{w_i} L \oslash \nabla_{b_i} L$ through gradients of unit i in the separation layer of the inference structure. Assume that $\max\left(1, \frac{\|\nabla_\omega L\|}{C}\right) = \Delta$, after gradients clipping, we have $\left(\frac{\nabla_{w_i} L}{\Delta}\right) \oslash \left(\frac{\nabla_{b_i} L}{\Delta}\right) = \nabla_{w_i} L \oslash \nabla_{b_i} L = x$. Clipping gradients without perturbation cannot prevent the proposed attack from reconstructing victims' samples. Adding noise to gradients without clipping also does little to defend against the proposed attack because the norm of gradients in the inference structure is significantly larger than the noise. Figure 8 compares the average $\ell 2$ norms and the absolute values of weight layer gradients under different conditions when the batch size is 16. The norm of the gradients in the inference structure is greater than the norm of the gradients generated by the regular model, and gradients in the proposed attack should be clipped in most cases. Figure 8(b) compares the absolute values of weight layer gradients with and without clipping. The perturbation is too small relative to the gradients without clipping, which has little effect on defending the proposed attack.

C. Performance Factors

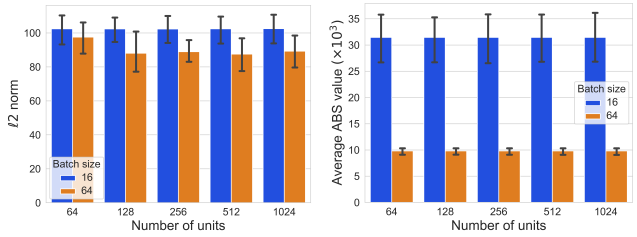
We discuss and analyze several factors that affect the performance of the proposed attack. First, we focus on the impact of user training models with different batch sizes on the quality of reconstructed samples. Figure 9 provides PSNR and CW-SSIM of reconstructed samples under different batch sizes where the number of the weight layer units in the inference structure is 2048. A larger batch size means that users' gradients contain more sample information. However, gradients are clipped by the clipping bound so that the $\ell 2$ norm of gradients does exceed the clipping bound. Therefore, a larger batch size reduces each sample's information in users' gradients, increasing the difficulty of reconstructing samples and reducing the quality. The evaluation shows that the proposed attack reconstructs samples with high quality when the batch size is small, a typical result of existing reconstruction attacks. Choosing a larger batch size in FL is a straightforward and effective way to defend against reconstruction attacks.

We next discuss the impact of the number of the weight layer units in the inference structure on the proposed attack. Figure 10 shows that the number of weight layer units has little impact on the quality of reconstructed images but a significant impact on gradient separation. The separation ratio refers to the proportion of separated reconstructed images, rather than overlapped images (such as Fig. 4), in a batch of training samples. Increasing the number of units is an effective way to avoid overlapped images without considering the gradient expansion. Figure 11 provides the $\ell 2$ norms and absolute values of the weight layer gradients under different numbers of units. As given in Theorem 3, the $\ell 2$ norm of gradients does not increase with the number of units because the number of units with non-zero gradients is not greater than the batch size. The number of batch sizes has the most significant impact



(a) Batch size = 16 (b) Batch size = 64

Fig. 10. The ratio of separated reconstructed samples in a batch of training samples.



(a) $\ell 2$ norm of gradients (b) Absolute value of gradients

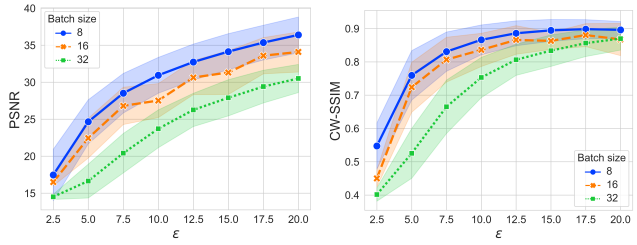
Fig. 11. The $\ell 2$ norms and absolute values of gradients without clipping under different numbers of the weight layer units in the inference structure.

on the absolute value of gradients. Given a clipping bound, a larger batch size means more unit has non-zero gradients, which reduces the absolute values of gradients in each unit.

D. Privacy Parameter Setting

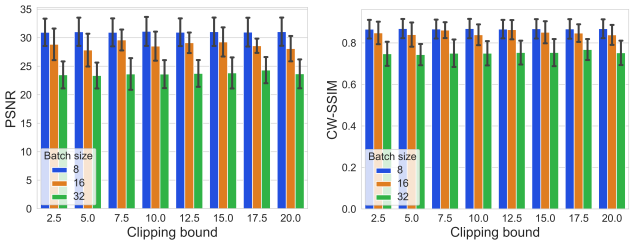
We discuss the impact of LDP privacy parameters on the performance of the proposed attack. The most direct impact on performance is the privacy parameter ε. As shown in Section VII-A, the scale of the Gaussian noise for gradient perturbation depends on ε. A larger ε means a smaller noise scale and less perturbation noise in general. Thus, as given in Fig. 12, a smaller ε provides better protection for gradients and reduces reconstructed sample quality.

We next discuss the impact of clipping bounds on the proposed attack with Fig. 13. Clipping bounds represent the compression degree of users' gradients. However, we find that a larger clipping bound does not improve the quality of reconstructed samples. The reason is that the scale of perturbation noise also depends on clipping bounds, and a larger clipping bound means more noise in users' gradients.



(a) PSNR (b) CW-SSIM

Fig. 12. The quality of reconstructed samples under various ε.



(a) PSNR (b) CW-SSIM

Fig. 13. The quality of reconstructed samples under various clipping bounds.

TABLE V
DIFFERENCE PROPORTION BETWEEN GRADIENTS WITH AND WITHOUT IMPLEMENTING THE PROPOSED ATTACK UNDER VARIOUS USERS.

Range \ Users	100	50	10	5	2
$(-\infty, 10^{-6})$	36%	36%	32%	19%	17%
$[10^{-6}, 10^{-5})$	23%	22%	17%	15%	7%
$[10^{-5}, 10^{-4})$	24%	23%	24%	23%	17%
$[10^{-4}, 10^{-3})$	16%	18%	22%	25%	23%
$[10^{-3}, 10^{-2})$	1%	1%	5%	18%	26%
$[10^{-2}, +\infty)$	0%	0%	0%	0%	10%

Therefore, increasing clipping bounds has little effect on improving reconstructed quality.

E. Impact of the Proposed Attack on FL Training

In this part, we explore the impact of the proposed attack on FL training. We first show the difference in the aggregated target model gradients between the two cases of whether or not the proposed attack is implemented. Then, we compared the accuracy of the final global model under two conditions to illustrate that the proposed attack has almost no impact on the performance of the target model.

Although the proposed attack only requires one training round of gradients to implement, we attack different victims in each round since a user's gradients in FL training have a limited impact on the final model. For example, when there are 50 users in each FL training round, and the number of training rounds is 10K, the server obtains a total of 500K users' gradients. The proposed attack on a single user only removes one gradient from 500K gradients. Therefore, we carry out the proposed attack in each training round, and the number of victims in the above example is 10K.

Table V compares the difference between aggregated gradients of the target model with and without implementing the proposed attack under various training users in each round. In the case of extremely few users (number of users is 2), whether or not the attack is implemented significantly impacts the aggregation gradient. In this case, the proposed attack causes aggregated gradients to be determined by only one user (the only non-target user); otherwise, the aggregate gradients are the average of two users. However, as the number of users increases, the gradient difference caused by the proposed attack becomes smaller. The number of training users can

easily exceed 100 in FL, and the proposed attack has almost no effect on the aggregated gradient.

The same phenomenon also appears in the model accuracy. We set the number of training users to 50 and calculate the top-1 accuracy (Acc@1) and top-5 accuracy (Acc@5) of various convergent target models ([51]–[54]) on the ImageNet [36], while the results are summarized in Table IX of the Appendix A. Training users' samples are random samples from the ImageNet training set. Even without perturbing gradients, the proposed inference structure has little impact on the performance of the target model when the number of training rounds is sufficient. When considering the noise in the gradient, the performance difference is more challenging to detect since LDP causes a certain degree of performance loss.

VIII. Discussion

A. Attack Consumption

The proposed attack only requires users to upload the gradient once, and samples can be reconstructed through gradients in a single training round. In other words, users do not repeatedly upload multiple rounds of gradients, which increases the stealth of the proposed attack. Although training users are randomly selected in each training round, the proposed attack can reconstruct victims' samples through one-round gradients. In addition, experiments in Section VII-E show that the proposed attack has little effect on the training and accuracy of the model when there is only one victim per training round. The size of gradients is related to model complexity. For a global model with parameters ω, the size of local gradients is also $|\omega|$. Since the victim only uploads one round of gradients in the proposed attack, the communication consumption of the proposed attack is $\mathcal{O}(|\omega|)$.

We discuss the computational cost of the proposed attack by providing the time complexity of each step in Section VI-B. Without loss of generality, we consider the batch size to be B and the size of samples to be S. The number of units in weight, bias, and metrics layers in the inference structure are N_w, N_b, and N_m, respectively. Step (1) infers the privacy parameter from the mean and variance of weight gradients in the separation layer, and the complexity is $\mathcal{O}(N_w S)$. Steps (2) and (3) calculate the bias gradient by mean and perform raw reconstruction by element-wised division, which are of complexity $\mathcal{O}(N_w N_b)$ and $\mathcal{O}(N_w S)$, respectively. Step (4) classifies the metrics according to their position, which is an in-place operation. Step (5) averages and reorders the metrics, and its complexity is $\mathcal{O}(n_m B) + \mathcal{O}(B \log B)$. Step (6) selects valid results by reconstructed reverse units, and its complexity is $\mathcal{O}(B)$. Step (7) is an optimization problem involving BS variables. The essence of step (8) is the comparison and assignment operation, and its complexity is $\mathcal{O}(BS)$.

TABLE VI provides the running time for performing a single proposed attack through the victim's clipped and perturbed gradients. The computational consumption of the proposed attack is concentrated in metric-based sample optimization. Batch sizes also significantly impact running time since the increase in training samples increases the number of optimization variables and expands the solution space.

TABLE VI
Running time (s) of the proposed attack under various batch sizes (B) and optimization rounds (R).

R \ B	4	8	16	32	64
0	0.032	0.034	0.036	0.037	0.039
500	1.049	1.101	1.157	1.389	1.483
1000	1.871	1.994	2.038	2.552	2.767
2000	3.558	4.298	4.299	4.641	5.645
4000	7.289	8.433	9.075	9.330	11.275

TABLE VII
Power consumption (W) of the proposed attack under various batch sizes (B) and optimization rounds (R).

R \ B	4	8	16	32	64
0	58.961	60.706	62.283	64.763	73.595
500	67.617	72.269	85.212	110.673	164.893
1000	68.403	75.083	90.305	123.168	195.431
2000	69.678	82.241	91.052	139.835	207.097
4000	73.760	84.106	94.220	140.651	214.178

Besides, Table VII records the average powers of the GPU when performing the proposed attack. Logging of power consumption data is implemented by PyTorch.[3] The number of optimization rounds has little effect on the average power, as the video memory consumption tends to stabilize during the optimization. The increase in optimization variables means more memory consumption. As a result, batch sizes also significantly impact the average power.

B. Possible Defense and attack limitations

Possible defenses against the proposed attacks include using dynamic clipping bounds and privacy parameters in local training, model malicious structure detection for users, and cryptography-based secure gradient aggregation (e.g., [49], [50]). In addition, we will consider attacks on model updates, design better optimization objectives, and expand to more data types in future work. Due to the length limitation of the paper, we discuss possible defenses and limitations of the proposed attack in more detail in Appendix B.

IX. Conclusion

This paper proposes a sample reconstruction attack against FL mechanisms with LDP, in which gradients are clipped and noisy. We briefly analyze the reason for the gradient expansion and the failure of the existing separation methods. Based on the above analysis, we design the proposed attack from two aspects: gradient separation without expansion and sample quality improvement against FL with LDP. We present a separation layer such that the gradients of each sample only exist in its reverse unit, which effectively separates

[3]https://pytorch.org/docs/stable/generated/torch.cuda.power_draw.html.

gradients in FL with LDP without causing gradient expansion. Besides, the subjects of samples selected by SAM can further compress victims' gradients. For sample quality improvement, we infer the confidence interval of the noise by the artificially added zero gradients and filter the noise in the background of reconstructed samples. In addition, a metric-based optimization is proposed to improve the sample quality further. Theory and evaluations show that the proposed attack is the only reconstruction attack that effectively reconstructs victims' samples when gradients are clipped and noisy. Simulation results show that the proposed attack has little impact on FL training and model accuracy. Finally, we provide possible defenses and discuss the limitations and future works.

REFERENCES

[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. A. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019.

[2] X. Mu, K. Cheng, Y. Shen, X. Li, Z. Chang, T. Zhang, and X. Ma, "Feddmc: Efficient and robust federated learning via detecting malicious clients," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[3] X. Mu, Y. Shen, K. Cheng, X. Geng, J. Fu, T. Zhang, and Z. Zhang, "Fedproc: Prototypical contrastive federated learning on non-iid data," *Future Generation Computer Systems*, vol. 143, pp. 93–104, 2023.

[4] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," 2017.

[5] M. Paulik, M. Seigel, H. Mason, D. Telaar, J. Kluivers, R. van Dalen, C. W. Lau, L. Carlson, F. Granqvist, C. Vandevelde, S. Agarwal, J. Freudiger, A. Byde, A. Bhowmick, G. Kapoor, S. Beaumont, A. Cahill, D. Hughes, O. Javidbakht, F. Dong, R. Rishi, and S. Hung, "Federated evaluation and tuning for on-device personalization: System design & and applications," 2021.

[6] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "Fate: An industrial grade platform for collaborative learning with data protection," *Journal of Machine Learning Research*, vol. 22, no. 226, pp. 1–6, 2021.

[7] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[8] H. Yang, M. Ge, K. Xiang, and J. Li, "Using highly compressed gradients in federated learning for data reconstruction attacks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 818–830, 2023.

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

[10] L. H. Fowl, J. Geiping, W. Czaja, M. Goldblum, and T. Goldstein, "Robbing the fed: Directly obtaining private data in federated learning with modified models," in *International Conference on Learning Representations*, 2022.

[11] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot, "When the curious abandon honesty: Federated learning is not private," 2021.

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

[13] D. Pasquini, D. Francati, and G. Ateniese, "Eluding secure aggregation in federated learning via model inconsistency," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, (New York, NY, USA), p. 2429–2443, Association for Computing Machinery, 2022.

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

[15] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot, "Reconstructing individual data points in federated learning hardened with differential privacy and secure aggregation," in *2023 IEEE 8th European Symposium on Security and Privacy*, (Los Alamitos, CA, USA), pp. 241–257, IEEE Computer Society, jul 2023.

[16] J. Jeon, K. Lee, S. Oh, J. Ok, *et al.*, "Gradient inversion with generative image prior," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29898–29908, 2021.

[17] A. Hatamizadeh, H. Yin, H. R. Roth, W. Li, J. Kautz, D. Xu, and P. Molchanov, "Gradvit: Gradient inversion of vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10021–10030, June 2022.

[18] M. Song, Z. Wang, Z. Zhang, Y. Song, Q. Wang, J. Ren, and H. Qi, "Analyzing user-level privacy attack against federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2430–2444, 2020.

[19] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, "Cafe: Catastrophic data leakage in vertical federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 994–1006, 2021.

[20] J. Zhu and M. Blaschko, "R-gap: Recursive gradient attack on privacy," 2021.

[21] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[22] J. Zhou, N. Wu, Y. Wang, S. Gu, Z. Cao, X. Dong, and K.-K. R. Choo, "A differentially private federated learning model against poisoning attacks in edge computing," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2022.

[23] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9530–9539, 2020.

[24] T. Stevens, C. Skalka, C. Vincent, J. Ring, S. Clark, and J. Near, "Efficient differentially private secure aggregation for federated learning via hardness of learning with errors," in *31st USENIX Security Symposium (USENIX Security 22)*, (Boston, MA), pp. 1379–1395, USENIX Association, Aug. 2022.

[25] X. Pan, M. Zhang, Y. Yan, J. Zhu, and Z. Yang, "Exploring the security boundary of data reconstruction via neuron exclusivity analysis," in *31st USENIX Security Symposium (USENIX Security 22)*, (Boston, MA), pp. 3989–4006, USENIX Association, Aug. 2022.

[26] M. Khosravy, K. Nakamura, Y. Hirose, N. Nitta, and N. Babaguchi, "Model inversion attack by integration of deep generative models: Privacy-sensitive face generation from a face recognition system," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 357–372, 2022.

[27] G. Ganev and E. D. Cristofaro, "On the inadequacy of similarity-based privacy metrics: Reconstruction attacks against "truly anonymous synthetic data"," 2023.

[28] X. Yuan, X. Ma, L. Zhang, Y. Fang, and D. Wu, "Beyond class-level privacy leakage: Breaking record-level privacy in federated learning," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2555–2565, 2021.

[29] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[30] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (A. Singh and J. Zhu, eds.), vol. 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, PMLR, 20–22 Apr 2017.

[31] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[32] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *International Conference on Learning Representations*, 2018.

[33] M. Naseri, J. Hayes, and E. De Cristofaro, "Local and central differential privacy for robustness and privacy in federated learning," 2022.

[34] D. Pasquini, D. Francati, and G. Ateniese, "Eluding secure aggregation in federated learning via model inconsistency," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2429–2443, 2022.

[35] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients - how easy is it to break privacy in federated learning?," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 16937–16947, Curran Associates, Inc., 2020.

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[37] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[38] C. Zhang, L. Liu, Y. Cui, G. Huang, W. Lin, Y. Yang, and Y. Hu, "A comprehensive survey on segment anything model for vision and beyond," *arXiv preprint arXiv:2305.08196*, 2023.

[39] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," 2023.

[40] Y. Jing, X. Wang, and D. Tao, "Segment anything in non-euclidean domains: Challenges and opportunities," 2023.

[41] K. Cooray and M. M. Ananda, "A generalization of the half-normal distribution with applications to lifetime data," *Communications in Statistics—Theory and Methods*, vol. 37, no. 9, pp. 1323–1337, 2008.

[42] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester, *A Modern Introduction to Probability and Statistics: Understanding why and how*, vol. 488. Springer, 2005.

[43] H. G. Hong, Y. Cho, H. Cho, J. Ahn, and J. Kim, "Foreseeing reconstruction quality of gradient inversion: An optimization perspective," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 12473–12481, 2024.

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

[45] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[46] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, IEEE, 2008.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[48] M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, and M. K. Markey, "Complex wavelet structural similarity: A new image similarity index," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2385–2401, 2009.

[49] T. Nguyen and M. T. Thai, "Preserving privacy and security in federated learning," *IEEE/ACM Transactions on Networking*, vol. 32, no. 1, pp. 833–843, 2024.

[50] A. Song, J. Fu, X. Mu, X. Zhu, and K. Cheng, "L-secnet: Towards secure and lightweight deep neural network inference," *Journal of Networking and Network Applications*, no. 4, pp. 171–181, 2023.

[51] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," 2022.

[52] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.

[53] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020.

[54] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

Zhichao You received his B.E. degree in Information and Computing Science from South China Agricultural University, Guangzhou, China in 2018, the M.S. degree in Computer Science and Technology in Xidian University, Xi'an, China in 2021. He is now pursuing his Ph.D. degree in computer science and technology at Xidian University , Xi'an, China. His research interests include privacy-preserving federated learning and wireless network security.

Xuewen Dong (Member, IEEE) received the B.E., M.S., and Ph.D. degrees in Computer Science and Technology from Xidian University, Xi'an, China, in 2003, 2006, and 2011, respectively. From 2016 to 2017, he was with Oklahoma State University, OK, USA, as a visiting scholar. Now, he is an associate professor in the School of Computer Science in Xidian University. His research interests include cognitive radio networks, wireless network security, and blockchain.

Shujun Li received the BE degree in Information Science and Engineering, and the PhD degree in Information and Communication Engineering from Xi'an Jiaotong University, China, in 1997 and 2003, respectively. Currently, he is Professor of Cyber Security and directing the Institute of Cyber Security for Society (iCSS) at the University of Kent, UK. His current research interests mainly focus on interplays between several interdisciplinary research areas, including cyber security and privacy, cybercrime and digital forensics, human factors, multimedia computing, AI and NLP, and more recently education. He is a Fellow of the BCS, The Chartered Institute for IT and a Vice President of the Association of British Chinese Professors (ABCP).

Ximeng Liu (Senior Member, IEEE) received the B.Sc. degree in electronic engineering and the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2010 and 2015, respectively. He is currently a Full Professor with the College of Mathematics and Computer Science, Fuzhou University, China. He was also a Research Fellow with the School of Information System, Singapore Management University, Singapore. His research interests include cloud security, applied cryptography, and big data security.

Siqi Ma (Member, IEEE) received the B.S. degree in computer science from Xidian University, Xi'an, China, in 2013, and the Ph.D. degree in information systems from Singapore Management University in 2018. She was a Research Fellow with the Distinguished System Security Group, CSIRO. She is currently a Senior Lecturer with the University of New South Wales, Canberra Campus, Australia. Her current research interests include data security, the IoT security, and software security.

Yulong Shen (Member, IEEE) received the B.S. and M.S. degrees in computer science and the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2002, 2005, and 2008, respectively. He is currently a Professor with the School of Computer Science and Technology, Xidian University, and also an Associate Director of the Shaanxi Key Laboratory of Network and System Security. His research interests include wireless network security and cloud computing security.