Concealing Sensitive Samples against Gradient Leakage in Federated Learning

Jing Wu[1], **Munawar Hayat**[2], **Mingyi Zhou**[3], **Mehrtash Harandi**[1]

[1]Department of Electrical and Computer Systems Engineering, Monash University
[2]Department of Data Science and AI, Monash University
[3]Department of Software Systems and Cybersecurity, Monash University
{jing.wu1, munawar.hayat, mingyi.zhou, mehrtash.harandi}@monash.edu

Abstract

Federated Learning (FL) is a distributed learning paradigm that enhances users' privacy by eliminating the need for clients to share raw, private data with the server. Despite the success, recent studies expose the vulnerability of FL to model inversion attacks, where adversaries reconstruct users' private data via eavesdropping on the shared gradient information. We hypothesize that a key factor in the success of such attacks is the low entanglement among gradients per data within the batch during stochastic optimization. This creates a vulnerability that an adversary can exploit to reconstruct the sensitive data. Building upon this insight, we present a simple, yet effective defense strategy that obfuscates the gradients of the sensitive data with concealed samples. To achieve this, we propose synthesizing concealed samples to mimic the sensitive data at the gradient level while ensuring their visual dissimilarity from the actual sensitive data. Compared to the previous art, our empirical evaluations suggest that the proposed technique provides the strongest protection while simultaneously maintaining the FL performance. Code is located at https://github.com/JingWu321/DCS-2.

Introduction

Consider an Artificial Intelligence (AI) service that aids in disease diagnosis. Multiple hospitals train a model for this service in collaboration. Publishing such a service could benefit a large number of doctors and patients, but it is critical to ensure that private medical data is secure and the utility of the service is normal. Federated Learning (FL) (McMahan et al. 2017a) is an essential technology for such critical applications where the confidentiality of private data is important. FL provides a distributed learning paradigm that enables multiple clients (*e.g.*, hospitals, businesses, or even mobile devices) to train a unified model jointly under the orchestration of a central server. A key advantage of FL lies in its promise of privacy for participating clients. With data decentralized and users' information kept solely with the client, only model updates (*e.g.*, gradients) are transmitted to the central server. Since the model's updates are specifically tailored to the learning task, they may create a false sense of security for FL clients, leading them to believe that the shared updates contain no information on their private training data (Kairouz et al. 2021).

Recent *model inversion attacks* (Zhu, Liu, and Han 2019; Geiping et al. 2020; Balunović et al. 2022; Fowl et al. 2022; Li et al. 2022) have shown that the users' private data can be reconstructed from the gradients shared during the learning process. This alarming finding has led to the exploration of various defense schemes to mitigate privacy leakage. Zhu *et al*. (Zhu, Liu, and Han 2019) employed a strategy that adds noise to gradients, guided by Differential Privacy (DP) (Dwork et al. 2006; Abadi et al. 2016; Song, Chaudhuri, and Sarwate 2013; McMahan et al. 2017b), a concept originally designed to constrain information disclosure. They also utilized gradient compression (Lin et al. 2017), which prunes gradients below a threshold magnitude, as a protective measure. Latest techniques have further advanced the field, with developments such as Automatic Transformation Search (ATS) (Gao et al. 2021) (augmenting data to hide sensitive information), PRivacy EnhanCing mODulE (PRECODE) (Scheliga, Mäder, and Seeland 2022) (use of bottleneck to hide the sensitive data), and Soteria (Sun et al. 2021) (pruning gradients in a single layer).

However, as defense techniques improve, **attacks evolve** as well. New findings, as highlighted by Balunović et al. (2022) and Li et al. (2022), indicate that modern defenses may be ineffective against more sophisticated attacks. For example, Balunović et al. (2022) show that an adversary can disregard the gradients pruned by Soteria and still reconstruct inputs, even without knowledge of the specific layers where pruning is applied. The vulnerability also extends to other defenses; data can be readily reconstructed in the initial communication rounds against the defense ATS (Balunović et al. 2022). In the case of the defense PRECODE, the mere presence of a single non-zero entry in the bias term can enable perfect reconstruction by adversaries (Balunović et al. 2022).

Most current defenses seek to protect all data equally, even if this results in a poor privacy-performance trade-off. In this work, we argue for a more realistic and practical setup where the focus should be given to the sensitive data (*e.g.*, personal data revealing racial or ethnic origin, political opinions, and religious beliefs as mentioned in European Union's General Data Protection Regulation (Voigt and Von dem Bussche 2017)). Consider a malignant skin lesion recognition system as an example. Skin images with tattoos that contain personal information demand extra attention than

images without such information. As such, preserving the former's privacy should be the algorithms' priority.

Exploring the underlying mechanism of model inversion attacks, we hypothesize that these attacks capitalize on the characteristic of relatively low entanglement among the gradients of data points during stochastic optimization. Building upon this understanding, we introduce a defense strategy that obfuscates the gradients of sensitive data using concealed samples. Formally, our goal is to ensure that an adversary is unable to reconstruct sensitive data while simultaneously preserving the performance of the FL system. To achieve this, we propose an algorithm that can adaptively synthesize concealed samples in lieu of sensitive data. We design the concealed points to have high gradient similarity with the sensitive data but visually disparate. For this purpose, our proposed defense has two main characteristics; **1) Enhancing the privacy of sensitive data.** Even though the gradients from the concealed data are similar to those of the sensitive data, inverting these gradients results in data points that are visually very different from the sensitive data. By obfuscating the gradients of the sensitive data with those of the concealed data, the reconstruction of sensitive information becomes confounded, which in turn leads to enhancing the privacy of sensitive data in FL. **2) Maintaining the FL performance.** The introduction of concealed data could potentially disrupt the learning process as it alters the gradient information. Our algorithm mitigates this by ensuring that the shared gradients, after the introduction of concealed data, align with the gradients of the original training samples, including sensitive data. This alignment is achieved through a gradient projection-based approach, preserving the learning capability of the FL system. Unlike existing defenses, our approach proposes a practical solution to enhance privacy in FL. It presents a significant challenge for an adversary to reconstruct the user-defined sensitive samples, all without sacrificing the overall performance of the FL system.

Our main contributions can be summarized as follows:

- We show that model inversion attacks predominantly exploit the characteristic of relatively low entanglement among gradients of samples during stochastic optimization. Based on this finding, we propose to adaptively synthesize concealed samples that obfuscate the gradients of sensitive data.
- The proposed approach crafts concealed samples that are adaptively learned to enhance privacy for sensitive data while simultaneously avoiding performance degradation.
- We thoroughly evaluate and compare our algorithm against various baselines (*e.g.*, injecting noise to the gradients as in the previous works (Sun et al. 2021; Gao et al. 2021; Zhu, Liu, and Han 2019)), and empirically observe that our algorithm consistently outperforms the current state-of-the-art defense methods.

Related Work

Model Inversion Attacks. Several model inversion attacks breach FL privacy by reconstructing the clients' data *e.g.*, (Zhu and Blaschko 2020; Fan et al. 2020; Zhu, Liu, and Han 2019; Yin et al. 2021; Jin et al. 2021; Jeon et al. 2021; Li et al. 2022; Takahashi, Liu, and Liu 2023; Nguyen et al. 2023). Deep Leakage from Gradients (DLG) (Zhu, Liu, and Han 2019) and its variants (Zhao, Mopuri, and Bilen 2020) employ an optimization-based technique to reconstruct private data from the given gradient updates. While the original algorithm (Zhu, Liu, and Han 2019) works best if the number of training samples in each batch is small, subsequent works (Geiping et al. 2020; Wei et al. 2020; Mo et al. 2021; Jeon et al. 2021; Yin et al. 2021) including Gradient Similarity (GS) (Geiping et al. 2020) and GradInversion attack (Yin et al. 2021) are able to reconstruct high-resolution images with larger batch sizes by incorporating stronger image priors. Jin et al. (2021) introduce catastrophic data leakage (CAFE) in vertical federated learning (VFL), showing improved data recovery quality in VFL. Balunović et al. (2022) formalize the gradient leakage problem within the Bayesian framework and demonstrate that the existing optimization-based attacks could be approximated as the optimal adversary with different assumptions on the input and gradients (*ie.*, the prior knowledge about the input and conditional probability of the gradient given the input). They further show that most existing defenses are not quite effective against stronger attacks once appropriate priors (*e.g.*, using generative adversarial networks (Li et al. 2022)) are incorporated to reconstruct data.

While aforementioned optimization-based model inversion attacks assume the server is honest-but-curious (Goldreich 2009), recent works (Fowl et al. 2022; Boenisch et al. 2021) introduce model modification attacks by a malicious server. Boenisch et al. (2021) apply trap weights to initialize the model with the goal of activating parts of its parameters, enabling perfect reconstruction within milliseconds. Similarly, Fowl et al. (2022) proposes the insertion of a tailored imprint module into the network structure. The imprinting module will store information exclusively about a specific subset of data points during the updates, and as a result, data can be recovered precisely and quickly, even when aggregated over large batches.

Privacy Preserving Defenses. Several approaches propose defense against model inversion attacks that breach users' privacy in FL. We can broadly categorize the existing defenses against model inversion attacks into four categories: gradient compression (Lin et al. 2017; Sun et al. 2021) and perturbation (Dwork et al. 2006; Abadi et al. 2016; Song, Chaudhuri, and Sarwate 2013), data encryption (Gao et al. 2021; Huang et al. 2020), architectural modifications (Scheliga, Mäder, and Seeland 2022), and secure aggregation via changing the communication and training protocol (Bonawitz et al. 2017; Mohassel and Zhang 2017; Lee et al. 2021; Wei et al. 2021) (not considered here). Zhu, Liu, and Han (2019) show that gradient compression can help, while Sun et al. (2021) propose Soteria, suggesting gradient pruning in a single layer as a defense strategy. Zhu, Liu, and Han (2019) also explore adding Gaussian or Laplacian noise guided by DP (Dwork et al. 2006; Abadi et al. 2016; Song, Chaudhuri, and Sarwate 2013; McMahan et al. 2017b) to prevent data being reconstructed. ATS relies on

heavy data augmentation on training images to hide sensitive information, while InstaHide (Huang et al. 2020, 2021) encrypts the private data with data from public datasets. Scheliga, Mäder, and Seeland (2022) introduce PRECODE, which inserts a bottleneck to hide the users' data. Despite these significant efforts to develop defense schemes against FL attacks, recent works highlight the vulnerabilities of existing defenses. For example, several studies show that DP requires a large number of participants in the training process to converge (Zhu, Liu, and Han 2019; Gao et al. 2021; Sun et al. 2021). Balunović et al. (2022) show that an adversary can get an almost perfect reconstruction after dropping the gradients pruned by Soteria. Balunović *et al*. (Balunović et al. 2022) also suggests that it is easy to reconstruct the data using the GS attack in the initial communication rounds against ATS, while Carlini et al. (2020) shows that the private data can be recovered from the encodings of InstaHide (Huang et al. 2020, 2021). For PRECODE, Balunović et al. (2022) demonstrate that an adversary can completely reconstruct the data with at least one non-zero entry in the bias. Further, strong defenses like Soteria can still be bypassed by the Generative Gradient Leakage (GGL) attack method (Li et al. 2022).

Methodology

In this section, we outline our proposed defense against model inversion attacks. We begin by introducing a basic FL framework, followed by an explanation of a simple reconstruction formulation that illustrates how model inversion attacks operate with shared gradient information. Subsequently, we describe how our proposed approach counters these attacks. Throughout the paper, we denote scalars by lowercase symbols, vectors by bold lowercase symbols, and matrices by bold uppercase symbols (*e.g.*, a, $\boldsymbol{a}$, and $\boldsymbol{A}$).

Federated Learning

Let $f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$ be a model with parameters $\boldsymbol{\theta}$, classifying inputs $\boldsymbol{x} \in \mathcal{X}$ to labels $\boldsymbol{y}$ in the label space $\mathcal{Y}$. In FL, we assume that there are C clients and a central server. The data $\mathcal{D}_c$ resides with the client c, and the server receives the gradient updates from the clients to update the model parameters $\boldsymbol{\theta}$ as

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim\mathcal{D}_c}[\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{X}), \boldsymbol{Y}; \boldsymbol{\theta})]. \tag{1}$$

In the t-th training round, each client c will compute the gradients $\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{X}), \boldsymbol{Y})$ over local training data and send it to the server. The server then updates the model parameters $\boldsymbol{\theta}^t$ using gradients from the selected $\tilde{C}$ clients:

$$\boldsymbol{\theta}^t = \boldsymbol{\theta}^{t-1} - \frac{\eta}{\tilde{C}} \sum_{c=1}^{\tilde{C}} \nabla_{\boldsymbol{\theta}^{t-1}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{X}), \boldsymbol{Y}; \boldsymbol{\theta}^{t-1}), \tag{2}$$

where η is the learning rate. The server propagates back the updated parameters $\boldsymbol{\theta}^t$ to each client, repeating the process until convergence. Even though the private training data never leaves the local clients, in the following, we show how an adversary can still reconstruct the data based on the shared gradients $\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{X}), \boldsymbol{Y})$ from client c in the t-th communication round.

Privacy Leakage

Individual Data Point Leakage. Without loss of generality, we consider the case of a network having only one fully connected layer, for which the forward pass is given by $\mathbb{R}^m \ni \boldsymbol{y} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$, where $\boldsymbol{W} \in \mathbb{R}^{n\times m}$ is the weight and $\boldsymbol{b} \in \mathbb{R}^m$ is the bias. Let $\mathcal{L}$ denote the objective to update the parameters, then the adversary reconstructs the input $\boldsymbol{x} \in \mathbb{R}^n$ by computing the gradients of the objective w.r.t. the weight and the bias:

$$\begin{aligned} \nabla_{\boldsymbol{W}}\mathcal{L} &= [\frac{\partial\mathcal{L}}{\partial y_1}\frac{\partial y_1}{\partial \boldsymbol{W}_{:1}}, \cdots, \frac{\partial\mathcal{L}}{\partial y_m}\frac{\partial y_m}{\partial \boldsymbol{W}_{:m}}], \\ \nabla_{\boldsymbol{b}}\mathcal{L} &= [\frac{\partial\mathcal{L}}{\partial y_1}, \cdots, \frac{\partial\mathcal{L}}{\partial y_m}]. \end{aligned} \tag{3}$$

Note that $\frac{\partial y_l}{\partial \boldsymbol{W}_{:l}} = \boldsymbol{x}$ for $1 \leq l \leq m$. Thus, we can perfectly reconstruct the input from the gradient information as $\boldsymbol{x}^* = {}^{\nabla_{\boldsymbol{W}_{:l}}\mathcal{L}}/_{\nabla_{\boldsymbol{b}_l}\mathcal{L}} = (\frac{\partial\mathcal{L}}{\partial y_l}\frac{\partial y_l}{\partial \boldsymbol{W}_{:l}})/\frac{\partial\mathcal{L}}{\partial y_l} = \boldsymbol{x}$, provided that at least one element of the gradient of the loss with respect to the bias is non-zero (*ie*., $\frac{\partial\mathcal{L}}{\partial y_l} \neq 0, 1 \leq l \leq m$).

Multiple Data Points Leakage. Let $\boldsymbol{x}_j,\ j \in [1, B], B > 1$ denotes samples of a mini-batch of size B. The gradient of the mini-batch is:

$$\begin{aligned} \nabla_{\boldsymbol{W}}\mathcal{L} &= \frac{1}{B}\sum_{j=1}^{B}[\frac{\partial\mathcal{L}}{\partial y_{1,j}}\frac{\partial y_{1,j}}{\partial \boldsymbol{W}_{:1}}, \cdots, \frac{\partial\mathcal{L}}{\partial y_{m,j}}\frac{\partial y_{m,j}}{\partial \boldsymbol{W}_{:m}}], \\ \nabla_{\boldsymbol{b}}\mathcal{L} &= \frac{1}{B}\sum_{j=1}^{B}[\frac{\partial\mathcal{L}}{\partial y_{1,j}}, \cdots, \frac{\partial\mathcal{L}}{\partial y_{m,j}}], \end{aligned} \tag{4}$$

which encapsulates a linear combination of all data points $\boldsymbol{x}_j$ in the mini-batch. Sun et al. (2021) observe that for data coming from different classes, the corresponding data representations tend to be embedded in different rows/columns of gradients. Suppose that within the mini-batch, only $\boldsymbol{x}_1$ belongs to class y_c $(1 \leq c \leq m)$, then the column c of the gradient in Eq. (4) will have

$$\frac{\sum_{j=1}^{B}\frac{\partial\mathcal{L}}{\partial y_{c,j}}\frac{\partial y_{c,j}}{\partial \boldsymbol{W}_{:c}}}{\sum_{j=1}^{B}\frac{\partial\mathcal{L}}{\partial y_{c,j}}} \approx \frac{\frac{\partial\mathcal{L}}{\partial y_{c,1}}\frac{\partial y_{c,1}}{\partial \boldsymbol{W}_{:c}}}{\frac{\partial\mathcal{L}}{\partial y_{c,1}}} = \boldsymbol{x}_1. \tag{5}$$

Due to this property, *ie*. relatively low entanglement among gradients per data points within a batch, the adversary can reconstruct the data in practice. Boenisch *et al*. (Boenisch et al. 2021) also observe that for a ReLU network, overparameterization can cause all but one training data in a mini-batch to have zero gradients, allowing the individual data point leakage in the mini-batch and the passive adversaries to obtain perfect reconstruction in various cases.

Optimization-based attacks aim to reconstruct data by minimizing the distance between the gradient of the input and that of the reconstruction. In contrast, model modification attacks utilize specific parameters with the goal of amplifying the leakage of individual data points (Boenisch et al. 2021) within the mini-batch or allowing portions of the gradient to contain information exclusive to a subset of data points (Fowl et al. 2022). It is important to note that neither

optimization-based attacks nor model modification attacks can precisely separate the gradient for individual data points. **This limitation in the attack algorithms is a vulnerability that we leverage in our approach to protect the data.**

Defense by Concealing Sensitive Samples (DCS^2)

Our objective is to protect sensitive data without modifying any FL settings (*e.g.*, model structure) and the sensitive data themselves, while minimizing the impact of the proposed defense on the model performance. Previously, we discussed that model inversion attacks reconstruct the inputs using the gradient information since the gradient encapsulates sufficient information about data samples to reconstruct them (see Eq. (4)). We note that while theoretically, attacks cannot precisely separate the gradient for each sample, they can be extremely successful in practice. Our key insight is to insert samples (referred to as concealed samples) to imitate the sensitive data on the gradient level while ensuring that these samples are visually dissimilar to the sensitive data. Our goal is to make it difficult or even impossible for the adversary to distinguish the gradient of the synthesized concealed samples from the gradient of the sensitive data.

Without loss of generality, assume that there is only one sensitive data point, denoted by $\boldsymbol{x}_s$. Our task is to construct the concealed sample $\tilde{\boldsymbol{x}}_c$ for this sensitive data to achieve the following goals as part of our defense strategy:

Goal-1: To protect sensitive data from model inversion attacks, we would like to maximize the dissimilarity between the concealed sample $\tilde{\boldsymbol{x}}_c$ and the sensitive sample $\boldsymbol{x}_s$, as measured by $\|\tilde{\boldsymbol{x}}_c - \boldsymbol{x}_s\|$. Simultaneously, we seek to minimize the similarity between the gradient of the concealed sample w.r.t. sensitive data. This is quantified by the cosine similarity between the gradient vectors, *ie*., $\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c), \tilde{\boldsymbol{y}}_c)$ and $\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_s), \boldsymbol{y}_s)$, while ensuring that the resulting latent representation is similar to the sensitive latent representation, *ie*., $\|f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c) - f_{\boldsymbol{\theta}}(\boldsymbol{x}_s)\| \leq \epsilon$.

Goal-2: To facilitate the server's ability to learn and enhance the FL model, we must ensure that the resulting gradient closely resembles the gradient of the batch without concealed samples. This can be achieved by satisfying $\langle \nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\{\boldsymbol{x}_s\} \cup \{\tilde{\boldsymbol{x}}_c\}), \{\boldsymbol{y}_s\} \cup \{\tilde{\boldsymbol{y}}_c\}), \nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_s), \boldsymbol{y}_s)\rangle > 0$.

To accomplish the aforementioned goals, our defense strategy consists of two phases: **1.** *synthesizing the concealed samples* and **2.** *gradient projection*, which we discuss below.

Synthesizing the Concealed Samples. To obtain concealed samples that are visually dissimilar to sensitive data but whose gradient is similar to the sensitive data, we would like to solve the following optimization problem:

$$\min_{\tilde{\boldsymbol{x}}_c} 1 - \frac{\left\langle \nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c), \tilde{\boldsymbol{y}}_c), \nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_s), \boldsymbol{y}_s) \right\rangle}{\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c), \tilde{\boldsymbol{y}}_c)\| \, \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_s), \boldsymbol{y}_s)\|} \tag{6}$$

$$\max_{\tilde{\boldsymbol{x}}_c} \|\tilde{\boldsymbol{x}}_c - \boldsymbol{x}_s\| \tag{7}$$

$$\text{s.t.} \left\| f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c) - f_{\boldsymbol{\theta}}(\boldsymbol{x}_s) \right\| \leq \epsilon. \tag{8}$$

We propose the following objective to achieve this

$$\mathcal{L}_{obj} = (1 - \frac{\langle \nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c), \tilde{\boldsymbol{y}}_c), \nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_s), \boldsymbol{y}_s)\rangle}{\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c), \tilde{\boldsymbol{y}}_c)\| \times \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_s), \boldsymbol{y}_s)\|}) + e^{-\lambda_x \|\tilde{\boldsymbol{x}}_c - \boldsymbol{x}_s\|} + \lambda_z (\frac{\|f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c) - f_{\boldsymbol{\theta}}(\boldsymbol{x}_s)\|}{\|f_{\boldsymbol{\theta}}(\boldsymbol{x}_s)\|} - \epsilon), \tag{9}$$

where λ_z and λ_x are hyperparameters to balance the different terms in the objective, ϵ controls the latent distance. The first term and third term target achieving Goal-1 by ensuring that the concealed sample is similar to the sensitive data at the gradient level, while the second term learns the concealed sample to be visually dissimilar to the sensitive data.

Remark. *The label corresponding to the concealed sample $\tilde{\boldsymbol{x}}_c$ is denoted by $\tilde{\boldsymbol{y}}_c$ in Eq.* (9)*. To obtain $\tilde{\boldsymbol{x}}_c$, we solve an optimization problem, starting from $\boldsymbol{x}_0$, which may be a sample different from $\boldsymbol{x}_s$. In such cases, we assign $\tilde{\boldsymbol{y}}_c$ with the label of $\boldsymbol{x}_0$,* ie.*, $\tilde{\boldsymbol{y}}_c = \boldsymbol{y}_0$. In our experiments, we show that $\tilde{\boldsymbol{x}}_c$ can be randomly initialized, and accordingly, we set $\tilde{\boldsymbol{y}}_c$ at random. Our empirical evaluations show that the proposed method works equally well under both conditions.*

Gradient Projection. Using Eq. (9), we can obtain the concealed sample $\boldsymbol{x}_c$. What we need to do next is to ensure that the gradient of the mini-batch augmented with the concealed sample is aligned with the gradient of the original mini-batch, as this way, the server can improve its model. This will be achieved via the gradient projection, but before delving into details of projection and inspired by the mixup regularization (Zhang et al. 2017), we propose an enhancement. Let $\boldsymbol{g}$ be the gradient of the original mini-batch $\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_s), \boldsymbol{y}_s)$. We obtain the gradient with the concealed sample as

$$\boldsymbol{g}_c \triangleq \nabla_{\boldsymbol{\theta}} \Big\{ \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_s), \boldsymbol{y}_s) + \lambda_g \mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c), \tilde{\boldsymbol{y}}_c) + (1 - \lambda_g)\mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c), \boldsymbol{y}_s) \Big\}, \tag{10}$$

where λ_g is a hyperparameter. Note that if $\lambda_g = 1$, we indeed attain the gradient of the mini-batch augmented by the concealed sample. However, including the gradient in the form $\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_c), \boldsymbol{y}_s)$ is empirically observed to be beneficial. Analysis can be found in § .

To align the resulting gradient $\boldsymbol{g}_c$ with the original gradient of the mini-batch $\boldsymbol{g}$, we opt for the technique developed in (Lopez-Paz and Ranzato 2017). This will ensure that the gradient sent to the server will improve the FL model. To this end, we compute the angle between the original gradient vector and the new gradient and check if it satisfies $\langle \boldsymbol{g}, \boldsymbol{g}_c \rangle \geq 0$. If the constraints is satisfied, the new gradient $\boldsymbol{g}_c$ behaves similarly to that of obtained from the mini-batch $\boldsymbol{x}_s$; otherwise, we project the new gradient $\boldsymbol{g}_c$ to the closest gradient $\hat{\boldsymbol{g}}_c$ according to:

$$\begin{aligned} \arg\min_{\hat{\boldsymbol{g}}_c} \quad & \frac{1}{2} \|\boldsymbol{g}_c - \hat{\boldsymbol{g}}_c\|_2^2, \\ s.t. \quad & \langle \boldsymbol{g}, \hat{\boldsymbol{g}}_c \rangle \geq 0. \end{aligned} \tag{11}$$

Algorithm 1: Defense by DCS^2.

procedure GRADIENT OBFUSCATION
  initialize the start point for constructing the concealed data $\tilde{\boldsymbol{x}}_c \leftarrow \boldsymbol{x}_0, \tilde{\boldsymbol{y}}_c \leftarrow \boldsymbol{y}_0$;
  get the concealed sample $\tilde{\boldsymbol{x}}_c \leftarrow Eq.\ (9)$;
  compute the new gradient $\boldsymbol{g}_c \leftarrow Eq.\ (10)$;
procedure GRADIENT PROJECTION
  get the gradient from the original batch $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_s), \boldsymbol{y}_s)$;
  if $\langle \boldsymbol{g}, \boldsymbol{g}_c \rangle < 0$ **then**
    get the solution $v^* \leftarrow Eq.\ (12)$;
    project the new gradient to the closest gradient $\hat{\boldsymbol{g}}_c = \boldsymbol{g}v^* + \boldsymbol{g}_c$.

To efficiently solve Eq. (11), we employ the Quadratic Programming (QP) with inequality constraints:

$$\begin{aligned} \arg\min_{v} \quad & \frac{1}{2}\boldsymbol{g}^\top \boldsymbol{g} v + \boldsymbol{g}_c^\top \boldsymbol{g} v, \\ \text{s.t.} \quad & v \geq 0. \end{aligned} \tag{12}$$

The projected gradient $\hat{\boldsymbol{g}}_c$ is given from the solution v^* in Eq. (12) as $\hat{\boldsymbol{g}}_c = \boldsymbol{g}v^* + \boldsymbol{g}_c$. The complete pseudocode for the algorithm is provided in Algorithm 1.

Experiments

In this section, we first describe our evaluation settings, followed by a comparison of our defense with existing defenses against model inversion attacks in FL, to answer the following research questions (RQs):

- **RQ1:** Can the proposed method DCS^2 effectively protect sensitive data against model inversion attacks in FL?
- **RQ2:** Is the proposed method DCS^2 capable of maintaining FL performance while providing protection?
- **RQ3:** How does the proposed method DCS^2 compare with existing defenses?
- **RQ4:** How does the proposed method DCS^2 perform when defending against adaptive attacks?
- **RQ5:** How does the proposed method DCS^2 perform when the starting point for generating concealing samples varies?

Additional details and results are available in the supplementary material at https://arxiv.org/pdf/2209.05724v2.pdf.

Experimental Setup

Attack Methods. We evaluate defenses against classical and state-of-the-art (SOTA) attacks in FL: the improved version of the classical Deep Leakage from Gradients (Zhu, Liu, and Han 2019) called *GS attack* (Geiping et al. 2020) that introduces image prior and uses cosine similarity as a distance metric to enhance reconstruction, and SOTA attack *GGL attack* that uses a Generative Adversarial Network (GAN) to learn prior knowledge from public datasets. We also include the recently proposed SOTA model modification attack *ie*. *Imprint attack* (Fowl et al. 2022). Furthermore, we provide an evaluation when the *adaptive attack* has strong prior knowledge about the private training data.

Defense Baselines. Following Sun et al. (2021); Gao et al. (2021), we compare DCS^2 with defenses including *DP-Gaussian* (adding Gaussian noise to gradients, following the implementation in (Sun et al. 2021; Gao et al. 2021)), and *Prune* (Gradient Compression) (Lin et al. 2017). We further compare against the recently proposed defense *Soteria* (Sun et al. 2021), which perturbs the representations.

Datasets and Models. We consider four datasets, namely MNIST (LeCun et al. 1998), CIFAR10 (Krizhevsky, Hinton et al. 2009), CelebFaces Attributes (CelebA) Dataset (Liu et al. 2015) with image resolution rescaled to 32×32 for a fair evaluation on GGL attack and TinyImageNet (Le and Yang 2015) with image resolution rescaled to 224×224. Being consistent with existing literature, we consider three model architectures i.e., LeNet (LeCun et al. 1998) for MNIST, ConvNet (with the same structure as in Soteria (Sun et al. 2021)) for CIFAR10 and CelebA, ResNet18 (He et al. 2016) for TinyImageNet.

Metrics. To quantify the quality of reconstructed images and compare them with the sensitive data, we use peak signal-to-noise ratio (PSNR) as used in the work (Balunović et al. 2022), and structural similarity index measure (SSIM) (Wang et al. 2004). Besides, we use the learned perceptual image patch similarity (LPIPS) metric (Zhang et al. 2018) for experiments on TinyImageNet. When measuring PSNR and SSIM, lower values indicate better performances. When it comes to LPIPS, a higher number indicates a better performance. We report classification accuracy values on the respective test sets (denoted as Acc_T) and the protected data (denoted as Acc_S) to measure the FL performance.

Privacy-Performance Trade-Off

We consider 100% of the training data in the target client as sensitive samples. The optimal conditions for an adversary to invert gradients are a batch size of one, a low image resolution, and an untrained target network.

Results on MNIST and CIFAR10. We first evaluate defenses against the GS attack on the MNIST and CIFAR10 datasets using models with randomly initialized weights. Results on Tab. 1 indicate that, compared with existing defenses, our proposed approach provides a better defense against the GS attack. Specifically, on MNIST, the defense baselines reduce the PSNR from 59.20 to ~ 10, while our defense can reduce the PSNR to around 8. On CIFAR10, our method reduces the SSIM to 0.17 when other defenses only reduce it to around 0.3. In terms of the FL performance, as shown in Tab. 1, our proposed defense method DCS^2 largely retains the performance compared with other defenses. Specifically, on MNIST, when most defense baseline drops the performance by about 1% on the sensitive data, our defense maintains the performance.

Results on CelebA and TinyImageNet. Further, we compare different defenses for more complex datasets, with larger capacity networks, on CelebA and TinyImageNet, to defend against stronger attacks. We use randomly initialized weights and use the attribute gender as the target label in CelebA to perform binary classification. A pre-trained

	MNIST				CIFAR10			
Defense	PSNR↓	SSIM↓	Acc_S↑	Acc_T↑	PSNR↓	SSIM↓	Acc_S↑	Acc_T↑
None	59.20±2.71	1.00±4.87	86.98±0.00	87.16±0.01	20.41±3.15	0.73±0.09	90.35±0.04	80.41±0.01
DP-Gaussian	35.38±2.44	0.83±0.07	85.94±0.00	86.91±0.01	12.34±1.34	0.28±0.06	77.19±0.18	79.65±0.04
Prune	14.13±2.29	0.37±0.06	85.94±0.00	86.91±0.00	11.26±1.75	0.22±0.06	77.80±0.32	79.51±0.08
Soteria	9.67±1.09	0.30±0.07	86.98±0.00	86.94±0.00	11.48±1.42	0.29±0.06	**84.70±0.32**	79.76±0.04
DCS^2 (Ours)	**7.84±2.56**	**0.17±0.09**	**86.98±0.00**	**86.98±0.01**	**8.04±1.10**	**0.15±0.05**	80.39±0.07	**79.79±0.03**

Table 1: Defenses against GS attack on MNIST and CIFAR10. Values are averaged.

	CelebA				TinyImageNet			
Defense	PSNR↓	SSIM↓	Acc_S↑	Acc_T↑	SSIM↓	LPIPS↑	Acc_S↑	Acc_T↑
None	19.92±2.18	0.75±0.07	100.0±0.00	93.79±0.07	1.00±0.00	0.00±0.00	73.94±1.21	66.41±0.02
DP-Gaussian	13.95±1.52	0.44±0.08	90.51±0.47	93.19±0.04	1.00±0.00	0.00±0.00	53.28±0.78	65.65±0.07
Prune	9.57±2.66	0.24±0.12	91.41±1.10	93.25±0.06	0.91±0.12	0.16±0.20	52.77±0.07	**65.73±0.20**
Soteria	8.89±2.63	0.24±0.11	100.0±0.00	93.86±0.01	1.00±0.00	0.00±0.00	41.84±1.14	52.06±1.47
DCS^2 (Ours)	**8.24±2.71**	**0.17±0.12**	**100.0±0.00**	**94.31±0.01**	**0.79±0.22**	**0.22±0.23**	**59.88±0.71**	65.68±0.05

Table 2: Defenses against GGL attack on CelebA and Imprint attack on TinyImageNet. Values are averaged.

λ_g	SSIM↓	LPIPS↑	Acc_S↑	Acc_T↑
0.5	0.80±0.20	0.22±0.21	**60.33±0.71**	**65.76±0.04**
0.7	0.79±0.22	0.22±0.23	59.88±0.71	65.68±0.05
1.0	**0.78±0.22**	**0.23±0.23**	58.54±0.46	65.24±0.21

Table 3: DCS^2 with different λ_g on TinyImageNet.

ResNet18 was applied for TinyImageNet. As shown in Table 2, our defense provides the best protection while competitively maintaining the original FL performance. Specifically, on CelebA, defending against the GGL attack, our method provides the best protection, and the FL performance is even improved while defenses DP-Gaussian and Prune drop by around 0.5% on the test set. On TinyImageNet, when defending against the Imprint attack, the defense Soteria cannot know where the adversary would insert the imprint module, so it cannot withstand the Imprint attack. While most defenses cannot provide protection, our defense method increases the LPIPS from 0.00 to 0.22.

Fig. 2 shows the example of reconstructions from different attacks with defenses on different datasets. The attacks could still recover some parts of the sensitive data with other defenses, while they fail with our proposed defense method. The training process on various datasets with different defenses is illustrated in Fig. 3. Training with these defenses typically results in convergence. However, in the case of Soteria on TinyImageNet, approximately 90% of the representations are perturbed, resulting in a convergence failure.

Tab. 3 presents the results for DCS^2 on TinyImageNet under varying values of λ_g. As λ_g increases, the protection for sensitive data improves. However, this leads to a reduction in the performance of the FL system.

PSNR↓ (None)	SSIM↓ (None)	PSNR↓ (DCS^2)	SSIM↓ (DCS^2)
59.22±2.71	1.00±4.77	7.87±2.44	0.18±0.09

Table 4: Defend against adaptive attacks.

Figure 1: (a) and (b) are visualization examples for Tab. 4 and Tab. 6 respectively. (Best viewed in color)

Comparison Against Adaptive Attacks

We compared the proposed defense method DCS^2 against two SOTA attacks: Imprint and GGL. Imprint modifies the architecture, and GGL uses a GAN to learn prior knowledge from public datasets. As per Gao et al. (2021), both these attacks are adaptive since the adversary "starts the reconstruction from an image with certain semantic information" or "designs attack techniques instead of optimizing the distance between the real and dummy gradients". Results in Tabs. 1 and 2 indicate that our defense provides the best protection with minimal drop in accuracy. For example, on TinyImageNet, our defense reduces the SSIM score from 1.0 to 0.79. In comparison, the defense Prune decreases it to approximately 0.9 and other defenses prove inadequate against this attack. The accuracy of the FL system using our defense on the sensitive data decreases by about 14%, whereas other defenses drop exceeding 20%.

Further, we design another strong attack where the adversary has strong prior knowledge and initializes the GS attack

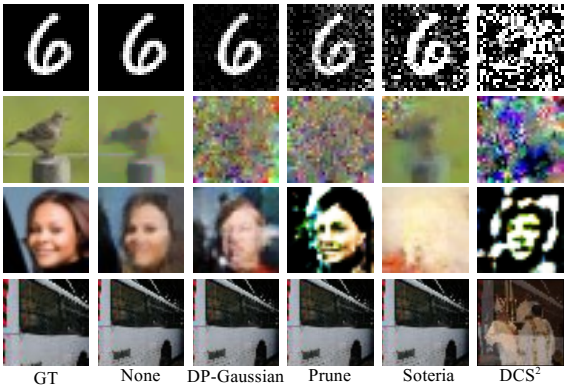

Figure 2: Example of reconstructions for Tabs. 1 and 2. From top to bottom are reconstructions from GS attacks on MNIST, those from GS attacks on CIFAR10, those from GGL attacks on CelebA, and those from Imprint attacks on TinyImageNet, respectively. (Best viewed in color)

MNIST	Noise	MixUP	PSNR↓	SSIM↓	Acc_S↑	Acc_T↑
✓	✗	✓	7.97	0.18	86.56	86.99
✓	✗	✗	7.84	0.17	86.98	86.98
✗	✓	✓	7.69	0.18	**86.98**	**86.99**
✗	✓	✗	**7.40**	**0.16**	85.94	86.94

Table 5: Different start points on MNIST.

with the average image for each class. Results are shown in Tab. 4, our proposed method can still provide good protection against such an attack with prior knowledge about the sensitive data. Fig. 1 (a) shows an example of the reconstructions from this attack. The GS attack would initialize the dummy input with the AvgImg (average image) shown in Fig. 1. The average image already explicitly includes information about the sensitive data, while our defense method could still protect the data against this adaptive attack.

Effect of Starting Points

We further evaluate our defense by choosing different initial starting points to craft the concealed samples. Tab. 5 show the performance with different start points. 'MixUP' means that $\tilde{\boldsymbol{x}}_c$ is initialized with $0.7\boldsymbol{x}_0 + 0.3\boldsymbol{x}_s$. Tab. 6 and Fig. 1 (b) show the results when the start points are from CIFAR10, which has different distribution than the target task dataset CelebA. As shown in Tabs. 5 and 6, even starting from random noise and different domains, our defense method could still provide protection and retain the model's performance.

Defense	PSNR↓	SSIM↓	Acc↑
None	19.92±2.18	0.75±0.07	93.79±0.07
DCS²	8.68±2.78	0.18±0.12	94.13±0.03

Table 6: Start points from CIFAR10 for CelebA.

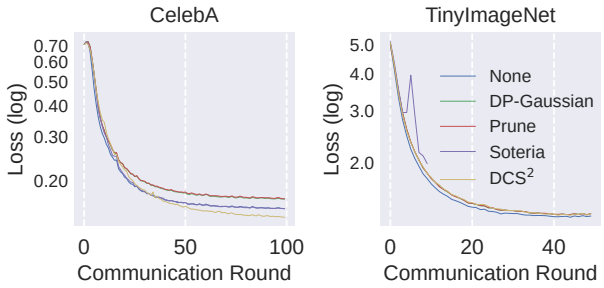

Figure 3: Training process. (Best viewed in color)

Defense	PSNR↓	SSIM↓	Acc_S↑	Acc_T↑
Prune	14.13	0.37	85.94	86.91
DCS²	7.84	0.17	**86.98**	**86.98**
Prune&DCS²	**6.08**	**0.12**	86.15	86.92

Table 7: Combination of defenses.

Combination With Existing Defenses

An illustration of combining DCS² with the defense 'Prune' is presented in Tab. 7. In this scenario, the enhancement of protection for private training data is notable. While the performance experiences a slight decrease compared to the standalone proposed defense method, it still surpasses the performance of the defense 'Prune' alone.

Limitations

While our empirical evaluations show that our proposed defense is effective in enhancing privacy and retaining FL performance, it requires additional computation to craft concealed samples (refer to the supplementary material for details on computation complexity). Future directions to improve concealed sample-based defense include finding the best starting points and reducing the time to craft concealed samples. We hope our defense can provide a new perspective for defending against model inversion attacks in FL.

Conclusion

In this work, we proposed an effective defense algorithm against model inversion attacks in FL. Our approach crafts concealed samples that imitate the sensitive data, but can obfuscate their gradients, thus making it challenging for an adversary to reconstruct sensitive data from the shared gradients. To enhance the privacy of the sensitive data, the concealed samples are adaptively learned to be visually very dissimilar to the sensitive samples, while their gradients are aligned with the original samples to avoid FL performance drop. Our evaluations on four benchmark datasets showed that, compared with other defenses, our approach offers the best protection against model inversion attacks while simultaneously retaining or even improving the FL performance.

Acknowledgments

Mehrtash Harandi gratefully acknowledges the support from the Australian Research Council (ARC), project DP230101176.

References

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

Balunović, M.; Dimitrov, D. I.; Staab, R.; and Vechev, M. 2022. Bayesian Framework for Gradient Leakage. *ICLR*.

Boenisch, F.; Dziedzic, A.; Schuster, R.; Shamsabadi, A. S.; Shumailov, I.; and Papernot, N. 2021. When the Curious Abandon Honesty: Federated Learning Is Not Private. *arXiv preprint arXiv:2112.02918*.

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

Carlini, N.; Deng, S.; Garg, S.; Jha, S.; Mahloujifar, S.; Mahmoody, M.; Song, S.; Thakurta, A.; and Tramer, F. 2020. Is Private Learning Possible with Instance Encoding? *arXiv preprint arXiv:2011.05315*.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 265–284. Springer.

Fan, L.; Ng, K. W.; Ju, C.; Zhang, T.; Liu, C.; Chan, C. S.; and Yang, Q. 2020. Rethinking privacy preserving deep learning: How to evaluate and thwart privacy attacks. In *Federated Learning*, 32–50. Springer.

Fowl, L.; Geiping, J.; Czaja, W.; Goldblum, M.; and Goldstein, T. 2022. Robbing the Fed: Directly Obtaining Private Data in Federated Learning with Modified Models. *ICLR*.

Gao, W.; Guo, S.; Zhang, T.; Qiu, H.; Wen, Y.; and Liu, Y. 2021. Privacy-preserving collaborative learning with automatic transformation search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 114–123.

Geiping, J.; Bauermeister, H.; Dröge, H.; and Moeller, M. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33: 16937–16947.

Goldreich, O. 2009. *Foundations of cryptography: volume 2, basic applications*. Cambridge university press.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

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

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

Jeon, J.; Lee, K.; Oh, S.; Ok, J.; et al. 2021. Gradient inversion with generative image prior. *Advances in Neural Information Processing Systems*, 34: 29898–29908.

Jin, X.; Chen, P.-Y.; Hsu, C.-Y.; Yu, C.-M.; and Chen, T. 2021. Catastrophic Data Leakage in Vertical Federated Learning. *Advances in Neural Information Processing Systems*, 34.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2): 1–210.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lee, H.; Kim, J.; Ahn, S.; Hussain, R.; Cho, S.; and Son, J. 2021. Digestive neural networks: A novel defense strategy against inference attacks in federated learning. *computers & security*, 109: 102378.

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

Lin, Y.; Han, S.; Mao, H.; Wang, Y.; and Dally, W. J. 2017. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.

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

McMahan, H. B.; Ramage, D.; Talwar, K.; and Zhang, L. 2017b. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*.

Mo, F.; Borovykh, A.; Malekzadeh, M.; Haddadi, H.; and Demetriou, S. 2021. Quantifying information leakage from gradients. *CoRR, abs/2105.13929*.

Mohassel, P.; and Zhang, Y. 2017. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, 19–38. IEEE.

Nguyen, N.-B.; Chandrasegaran, K.; Abdollahzadeh, M.; and Cheung, N.-M. 2023. Re-thinking Model Inversion Attacks Against Deep Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16384–16393.

Scheliga, D.; Mäder, P.; and Seeland, M. 2022. PRECODE-A Generic Model Extension to Prevent Deep Gradient Leakage. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1849–1858.

Song, S.; Chaudhuri, K.; and Sarwate, A. D. 2013. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, 245–248. IEEE.

Sun, J.; Li, A.; Wang, B.; Yang, H.; Li, H.; and Chen, Y. 2021. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9311–9319.

Takahashi, H.; Liu, J.; and Liu, Y. 2023. Breaching FedMD: Image Recovery via Paired-Logits Inversion Attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12198–12207.

Voigt, P.; and Von dem Bussche, A. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676): 10–5555.

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

Wei, W.; Liu, L.; Loper, M.; Chow, K.-H.; Gursoy, M. E.; Truex, S.; and Wu, Y. 2020. A framework for evaluating client privacy leakages in federated learning. In *European Symposium on Research in Computer Security*, 545–566. Springer.

Wei, W.; Liu, L.; Wut, Y.; Su, G.; and Iyengar, A. 2021. Gradient-leakage resilient federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, 797–807. IEEE.

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

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

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

Zhu, J.; and Blaschko, M. 2020. R-gap: Recursive gradient attack on privacy. *arXiv preprint arXiv:2010.07733*.

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