

CONCEALING SENSITIVE SAMPLES FOR ENHANCED PRIVACY IN FEDERATED LEARNING

Anonymous authors

Paper under double-blind review

ABSTRACT

Federated Learning (FL) is a distributed learning paradigm that promises to protect users' privacy by not requiring the clients to share their raw and private data with the server. Despite the success, recent studies reveal the vulnerability of FL to model inversion attacks by showing that they can reconstruct users' private data via eavesdropping on the shared gradient information. Most existing defence methods to preserve privacy in FL are formulated to protect *all* data samples equally, which in turn proven brittle against attacks and compromising the FL performance. In this paper, we argue that data containing sensitive information should take precedence. We present a simple, yet effective defence strategy that obfuscates the gradients of the sensitive data with concealed samples. In doing so, we propose to synthesize concealed samples to simulate the sensitive data at the gradient level. Furthermore, we employ a gradient projection technique to obscure sensitive data without compromising the quality of the shared gradients, hence enabling FL to retain its performance. Compared to the previous art, our empirical evaluations suggest that the proposed technique provides the strongest protection while simultaneously maintaining the FL performance. We also provide examples of how the proposed method can be combined with other defences to boost the privacy-performance trade-off even further.

1 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 *ie.* a misdiagnosis or underdiagnosis could have serious consequences. 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 is that it offers the promise of privacy to participating clients since the data is decentralized, the raw user data never leaves the client, and only model updates (*e.g.*, gradients) are communicated to the central server. Since the model updates are focused on the learning task, it provides a sense of security to the FL clients that the shared updates contain no information on their private data (Kairouz et al., 2021).

Even though the clients in FL never share their data with the server, recent *model inversion attacks* (Zhu et al., 2019; Geiping et al., 2020; Balunović et al., 2021; Fowl et al., 2021; Li et al., 2022) have shown that the users' private data can be reconstructed from the shared gradients. To address this privacy leakage, several defence schemes have been explored (*e.g.*, Differential Privacy (DP) by adding noise to gradients (Geyer et al., 2017), pruning gradients via compression (Lin et al., 2017)). Unfortunately, these defences suffer from noticeable performance degradation, as shown for example in (Zhu et al., 2019; Sun et al., 2021). Latest techniques such as Automatic Transformation Search (ATS) (Gao et al., 2021) (augmenting data to hide sensitive information), PRivacy Enhancing mODule (PRECODE) (Scheliga et al., 2022) (use of bottleneck to hide the sensitive data), and Soteria (Sun et al., 2021) (pruning gradients in a single layer) were initially shown to maintain the FL performance and simultaneously preserve the privacy. However, as defence techniques improve, attacks evolve as well. Recent studies such as Balunović et al. (2021); Li et al. (2022) raise the alarm by providing evidence that modern defences are ineffective against stronger attacks. For example,

Balunović et al. (2021) show that the adversary can drop the gradients pruned by the defence Soteria and still reconstruct inputs, without knowing to which layers pruning is applied to. Similarly, it is easy to reconstruct the data in the initial communication rounds against the defence ATS (Balunović et al., 2021). For the defence PRECODE, Balunović et al. (2021) show that the existence of at least one non-zero entry in the bias term can result in the perfect reconstruction of data by the adversary.

Most current defences seek to protect all data samples, 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 sensitive data (e.g., personal data revealing racial or ethnic origin, political opinions, religious beliefs¹). Taking a malignant skin lesion recognition system as an example, skin images with tattoos that contain personal information require extra focus than images without such information. As such, preserving the privacy of the former should be the priority of the algorithms.

Our goal is to enhance the privacy of the user’s defined important data. To ensure that an adversary is unable to reconstruct sensitive data, while the performance of the FL system is simultaneously maintained, we propose an algorithm that can adaptively synthesize concealed samples in lieu of the sensitive data. Specifically, we generate concealed points that have high gradient similarity with the sensitive data, while they are visually dissimilar with the sensitive data. For this purpose, our proposed defence has two main characteristics. **1) Enhancing the privacy of the 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. This is because the gradients from the sensitive data are obfuscated with those of the concealed data; thereby, reconstructing the sensitive data will be obfuscated with the concealed data, which in turn leads to enhancing the privacy of sensitive data in FL. **2) Maintaining the FL performance.** Introducing concealed data can potentially impact the learning process, as it alters the gradient information. Our algorithm ensures that the shared gradients (after introducing the concealed data) are aligned with the gradient of the original training samples (including sensitive data) by the proposed gradient projection based approach, thus maintaining the learning capability of the FL system. In contrast to existing defences, our approach therefore proposes a practical solution to privacy in FL, where it is a challenge for an adversary to reconstruct the user-defined sensitive samples, without compromising the overall performance of the FL system.

Contributions Our main contributions can be summarized as: **1.** The proposed approach crafts concealed samples, that is adaptively learned to enhance privacy for sensitive data while simultaneously avoiding performance degradation. Even when we are required to protect a large portion of sensitive data, the proposed algorithm can still improve privacy protection. **2.** Our algorithm is quite flexible and can be combined with existing defences (e.g., differential privacy) for achieving a desirable privacy-performance trade-off for all data. **3.** We thoroughly evaluate and compare our algorithm against various baselines (e.g., gradient compression), and empirically observe that our algorithm consistently outperforms the current state-of-the-art defence methods.

2 METHODOLOGY

In this section, we provide details of our proposed defence method against model inversion attacks. We first introduce a basic FL framework and discuss a simple reconstruction formulation to show how model inversion attacks work based on the shared gradients, and then describe how our proposed approach defends against these attacks. Throughout the paper, we denote scalars, vectors/matrices by lowercase and bold symbols, respectively (e.g., a , \mathbf{a} , and \mathbf{A}).

2.1 FEDERATED LEARNING

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

$$\min_{\theta} \frac{1}{C} \sum_{c=1}^C \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{D}_c} [\mathcal{L}(f_{\theta}(\mathbf{X}_c), \mathbf{Y}; \theta)]. \quad (1)$$

¹European Union’s General Data Protection Regulation (GDPR)

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

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

where η is the learning rate. The server propagates back the updated parameters θ^t to each client, and the process is repeated till 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 $\nabla_{\theta} \mathcal{L}(f_{\theta}(\mathbf{X}_c), \mathbf{Y}_c)$ from client c in the t -th communication round.

Remark. *If we assume that each client has its own objective, the FL problem can be formulated as*

$$\min_{\theta} \frac{1}{C} \sum_{c=1}^C \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_c} [\mathcal{L}_c(f_{\theta}(\mathbf{X}_c), \mathbf{Y}; \theta)].$$

Our solution is generic and can be used to address this problem as well.

2.2 RECONSTRUCTION FORMULATION

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 $\mathbb{R}^m \ni \mathbf{y} = \mathbf{W}^{\top} \mathbf{x} + \mathbf{b}$, where $\mathbf{W} \in \mathbb{R}^{n \times m}$ is the weight and $\mathbf{b} \in \mathbb{R}^m$ is the bias. Let \mathcal{L} denote the objective to update the parameters, then the adversary reconstructs the input $\mathbf{x} \in \mathbb{R}^n$ by computing the gradients of the objective w.r.t. the weight and the bias:

$$\begin{aligned} \nabla_{\mathbf{W}} \mathcal{L} &= \left[\frac{\partial \mathcal{L}}{\partial y_1} \frac{\partial y_1}{\partial \mathbf{W}_{:1}}, \dots, \frac{\partial \mathcal{L}}{\partial y_m} \frac{\partial y_m}{\partial \mathbf{W}_{:m}} \right], \\ \nabla_{\mathbf{b}} \mathcal{L} &= \left[\frac{\partial \mathcal{L}}{\partial y_1}, \dots, \frac{\partial \mathcal{L}}{\partial y_m} \right]. \end{aligned} \quad (3)$$

Note that $\frac{\partial y_l}{\partial \mathbf{W}_{:l}} = \mathbf{x}$ for $1 \leq l \leq m$. Thus, we can perfectly reconstruct the input from the gradient information as $\mathbf{x}^* = \nabla_{\mathbf{W}_{:l}} \mathcal{L} / \nabla_{\mathbf{b}_{:l}} \mathcal{L} = (\frac{\partial \mathcal{L}}{\partial y_l} \frac{\partial y_l}{\partial \mathbf{W}_{:l}}) / \frac{\partial \mathcal{L}}{\partial y_l} = \mathbf{x}$ if only one of the elements of the gradient of the loss w.r.t. the bias is non-zero (ie., $\partial \mathcal{L} / \partial y_l \neq 0, 1 \leq l \leq m$).

Multiple data points leakage Let $\mathbf{x}_j, j \in [1, B], B > 1$ denotes samples of a mini-batch of size B . The gradient of the mini-batch is:

$$\begin{aligned} \nabla_{\mathbf{W}} \mathcal{L} &= \frac{1}{B} \sum_{j=1}^B \left[\frac{\partial \mathcal{L}}{\partial y_{1,j}} \frac{\partial y_{1,j}}{\partial \mathbf{W}_{:1}}, \dots, \frac{\partial \mathcal{L}}{\partial y_{m,j}} \frac{\partial y_{m,j}}{\partial \mathbf{W}_{:m}} \right], \\ \nabla_{\mathbf{b}} \mathcal{L} &= \frac{1}{B} \sum_{j=1}^B \left[\frac{\partial \mathcal{L}}{\partial y_{1,j}}, \dots, \frac{\partial \mathcal{L}}{\partial y_{m,j}} \right], \end{aligned} \quad (4)$$

which encapsulates a linear combination of all data points \mathbf{x}_j in the mini-batch.

Boenisch et al. (2021) observe that for a ReLU network, over-parameterization 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 a perfect reconstruction in some cases. While optimization-based attacks seek to compute the reconstruction via minimizing the distance between the gradient of the input and that of the reconstruction, model modification attacks utilize particular parameters aiming to amplify the individual data point leakage (Boenisch et al., 2021) in the mini-batch, or allow portions of the gradient to only contain information of a subset data points (Fowl et al., 2021).

However, neither the optimization-based attack nor the model modification attack can precisely separate the gradient per data. This is a weakness of the attack algorithms, and we exploit this feature to protect the sensitive data.

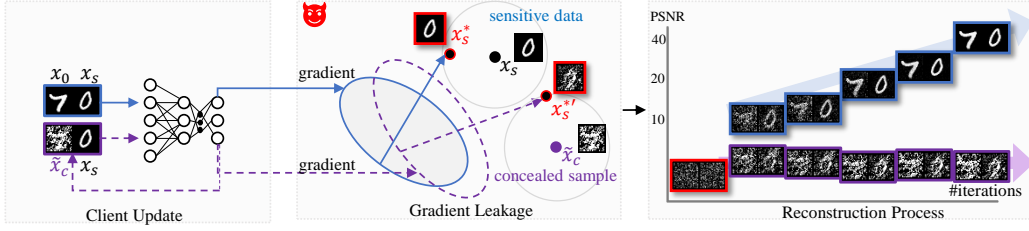


Figure 1: Illustration of gradient leakage and our defence. The adversary can obtain a perfect reconstruction x_s^* for the sensitive data x_s when the gradient is over the input x_0 and x_s , but it fails and get the reconstruction $x_s^{*'}$ when the gradient is over our concealed sample x_c and x_s .

2.3 DEFENCE BY CONCEALING SENSITIVE SAMPLES (DCS²)

Our objective is to protect sensitive data without modifying any FL system settings (e.g., model structure) and the sensitive data themselves, while minimizing the impact of the proposed defence on the model performance. Previously we discuss that model inversion attacks reconstruct the inputs using the gradient information since the gradient encapsulates a linear combination of data samples itself (see Eq. (4)). We note that while theoretically attacks cannot precisely separate the gradient for each data sample, in practice they can be extremely successful. Our key insight is to insert samples (referred to as concealed samples) to imitate the sensitive data on the gradient level in the mini-batch, 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 synthesised concealed samples from the gradient of the sensitive data.

Consider a sensitive data point x_s (see Figure 1 for an illustration), our aim is to craft the concealed sample \tilde{x}_c which makes $\nabla_{\theta} \mathcal{L}(f_{\theta}(\tilde{x}_c), \tilde{y}_c)$ approaching $\nabla_{\theta} \mathcal{L}(f_{\theta}(x_s), y_s)$. This strategy obfuscates the sensitive samples, as the reconstruction by the adversary through the gradient will contain information from the concealed sample, which we aim to be visually different from the sensitive data.

For the sake of discussion and without loss of generality, assume only one sensitive data x_s exists in the mini-batch \mathbf{X} . Our task is to construct the concealed sample \tilde{x}_c for the sensitive data to achieve the following goals as part of our defence:

- Goal-1:** We would like to maximize the dissimilarity between the concealed sample \tilde{x}_c and the sensitive sample x_s measured by $\|\tilde{x}_c - x_s\|$.
- Goal-2:** We also want to minimize the similarity between the gradient of the concealed sample w.r.t. sensitive data, measured by the cosine similarity between the gradient vectors, i.e., $\nabla_{\theta} \mathcal{L}(f_{\theta}(\tilde{x}_c), \tilde{y}_c)$ and $\nabla_{\theta} \mathcal{L}(f_{\theta}(x_s), y_s)$.
- Goal-3:** To enable the server to improve the learned FL model, we need to make sure that the resulting gradient resembles the gradient of the batch without concealed samples. This can be achieved by ensuring $\langle \nabla_{\theta} \mathcal{L}(f_{\theta}(\mathbf{X} \cup \{\tilde{x}_c\}), \mathbf{Y} \cup \{\tilde{y}_c\}), \nabla_{\theta} \mathcal{L}(f_{\theta}(\mathbf{X}), \mathbf{Y}) \rangle > 0$.

To accomplish all of the aforementioned goals, our defence 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 solve the following optimization problem:

$$\mathcal{L}_{obj} = -\frac{\langle \nabla_{\theta} \mathcal{L}(f_{\theta}(\tilde{x}_c), \tilde{y}_c), \nabla_{\theta} \mathcal{L}(f_{\theta}(x_s), y_s) \rangle}{\|\nabla_{\theta} \mathcal{L}(f_{\theta}(\tilde{x}_c), \tilde{y}_c)\| \times \|\nabla_{\theta} \mathcal{L}(f_{\theta}(x_s), y_s)\|} + \frac{\alpha}{\|\tilde{x}_c - x_s\|}, \quad (5)$$

where α is a hyperparameter to balance the two terms in the objective. In Eq. (5), the first term targets achieving Goal-2 by ensuring that the concealed sample is similar to the sensitive data at the gradient level, while the second term achieves Goal-1 and learns the concealed sample to be visually dissimilar to the sensitive data.

To further enhance the defence capability against the model modification attack (Fowl et al., 2021) which associates the gradients to a few samples, we add an extra constraint to bound the distance between the logits from the concealed sample and the sensitive data: $\|f_{\theta}(\tilde{x}_c) - f_{\theta}(x_s)\| \leq \epsilon$, where ϵ controls the latent distance. This constraint reduces the likelihood that a single data point will activate a single line in Eq. (4). So the final objective in Eq. (5) becomes:

$$\mathcal{L}'_{obj} = \mathcal{L}_{obj} + \beta \|f_{\theta}(\tilde{x}_c) - f_{\theta}(x_s)\|. \quad (6)$$

Here, β is a hyperparameter to control the contribution of various terms in the objective.

Remark. The label of the concealed sample \tilde{x}_c is denoted by \tilde{y}_c in Eq. (5). To obtain \tilde{x}_c , we solve an optimization problem, starting from x_0 that can be a sample different from x_s in the same mini-batch, ie., $x_0 \in \mathbf{X} \setminus \{x_s\}$. In that case, we assign \tilde{y}_c with the label of x_0 , ie., $\tilde{y}_c = y_0$. In our experiments, we show that \tilde{x}_c can be randomly initialized and therein, we set \tilde{y}_c randomly. Our empirical evaluations in § 4 show that the proposed method works equally well in both cases.

Gradient projection. Using Eq. (6), we can obtain the concealed sample 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 (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 \mathbf{g} be the gradient of the original mini-batch $\nabla_{\theta} \mathcal{L}(f_{\theta}(\mathbf{X}), \mathbf{Y})$. We obtain the gradient with the concealed sample as

$$\mathbf{g}_c = \mathbf{g} + \lambda \nabla_{\theta} \mathcal{L}(f_{\theta}(\tilde{x}_c), \tilde{y}_c) + (1 - \lambda) \nabla_{\theta} \mathcal{L}(f_{\theta}(\tilde{x}_c), y_s), \quad (7)$$

where λ is a hyperparameter². Note that if $\lambda = 1$, we indeed attain the gradient of the mini-batch augmented by the concealed sample. However, including the gradient in the form $\nabla_{\theta} \mathcal{L}(f_{\theta}(\tilde{x}_c), y_s)$ is empirically observed to be beneficial. The underlying idea is to enhance the gradient by making sure that the concealed sample is triggering the class of the sensitive sample as well.

To align the resulting gradient \mathbf{g}_c with the original gradient of the mini-batch \mathbf{g} , we opt for the technique developed in Lopez-Paz & 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 $\langle \mathbf{g}, \mathbf{g}_c \rangle \geq 0$. If the constraints is satisfied, the new gradient \mathbf{g}_c behaves similarly to that of obtained from the mini-batch \mathbf{X} ; otherwise, we project the new gradient \mathbf{g}_c to the closest gradient $\hat{\mathbf{g}}_c$ according to:

$$\begin{aligned} \arg \min_{\hat{\mathbf{g}}_c} \quad & \frac{1}{2} \|\mathbf{g}_c - \hat{\mathbf{g}}_c\|_2^2, \\ \text{s.t.} \quad & \langle \mathbf{g}, \hat{\mathbf{g}}_c \rangle \geq 0. \end{aligned} \quad (8)$$

To efficiently solve Eq. (8), we employ the Quadratic Programming (QP) with inequality constraints:

$$\begin{aligned} \arg \min_v \quad & \frac{1}{2} \mathbf{g}^{\top} \mathbf{g} v + \mathbf{g}_c^{\top} \mathbf{g} v, \\ \text{s.t.} \quad & v \geq 0. \end{aligned} \quad (9)$$

The projected gradient $\hat{\mathbf{g}}_c$ is given from the solution v^* in Eq. (9) as $\hat{\mathbf{g}}_c = \mathbf{g} v^* + \mathbf{g}_c$.

3 RELATED WORK

Model Inversion Attacks. Several model inversion attacks breach FL privacy by reconstructing the clients data e.g., (Zhu & Blaschko, 2020; Fan et al., 2020; Zhu et al., 2019; Yin et al., 2021; Jin et al., 2021; Jeon et al., 2021; Li et al., 2022). Deep Leakage from Gradients (DLG) (Zhu et al., 2019) and its variants (Zhao et al., 2020) employ an optimization-based technique to reconstruct private data from the given gradient updates. While the original algorithm Zhu et al. (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 using cosine similarity as the distance metric and incorporating stronger

²We set this to 0.3 for all experiments in § 4.

image priors. Balunović et al. (2021) 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. They further show that most existing defences are not quite effective against stronger attacks, once stronger priors (e.g., using generative adversarial networks (Li et al., 2022)) are incorporated to reconstruct data.

While above mentioned attacks assume the server is honest-but-curious (Goldreich, 2009), recent works Fowl et al. (2021); 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 single row with a single data point, enabling perfect reconstruction within milliseconds. Similarly, Fowl et al. (2021) propose Imprint attack to insert the imprint module with specific weights into the structure, allowing portions of the updates to only contain information about a subset of data points, which could recover data precisely and quickly, even if the data is aggregated over large batches.

Privacy Preserving Defences. Several approaches have been proposed in the literature to defend against FL attacks that breach users’ privacy. We can broadly categorize the existing defences against model inversion attacks into four categories: gradient compression (Lin et al., 2017; Sun et al., 2021) and perturbation (Geyer et al., 2017; McMahan et al., 2017b), data encryption (Gao et al., 2021; Huang et al., 2020), architectural modifications (Scheliga et al., 2022), and secure aggregation via changing the communication and training protocol (Bonawitz et al., 2017; Mohassel & Zhang, 2017; Lee et al., 2021; Wei et al., 2021) (not considered here). Zhu et al. (2019) show that gradient compression can help, while Sun et al. (2021) propose Soteria, which shows gradient pruning in a single layer as a defence strategy. Differential Privacy (DP) (Geyer et al., 2017; McMahan et al., 2017b) adds Gaussian or Laplacian noise into the gradients to prevent data being reconstructed. Automatic Transformation Search (ATS) relies on heavy data augmentation on training images to hide sensitive information. Scheliga et al. (2022) introduce a PRivacy Enhancing mODule (PRECODE), which inserts a bottleneck to hide the users’ data. Despite these significant efforts to develop defence schemes against FL attacks, recent works highlight the vulnerabilities of currently existing defences. For example, (Zhu et al., 2019; Gao et al., 2021; Sun et al., 2021) show that DP requires a large number of participants in the training process to converge. Balunović et al. (2021) show that an adversary can get an almost perfect reconstruction after dropping the gradients pruned by Soteria. Balunović et al. (2021) also suggests that its easy to reconstruct the data using the GS attack in the initial communication rounds against the defence ATS, while Carlini et al. (2020) shows that the private data can be recovered from the encodings of InstaHide. For the defence PRECODE, Balunović et al. (2021) demonstrate that there is always at least one non-zero entry in the bias, allowing the adversary to perfectly reconstruct the data. Further, strong defences like Soteria can still be bypassed by the Generative Gradient Leakage (GGL) attack method Li et al. (2022).

4 EXPERIMENTS

In this section, we first describe our evaluation settings, followed by a comparison of our defence with existing defences against model inversion attacks in FL. We evaluate the FL performance with defence techniques on Independent and Identically Distributed (IID) and Not Independent and Identically Distributed (Non-IID) settings. We further show that our proposed defence can complement existing defences and provide enhanced privacy protection for all data. Finally, we extensively analyze the proposed defence with different starting points for computing the concealed samples. More results and details can be found in Appendix A and B. Our source code is located in the repository.

4.1 EXPERIMENTAL SETUP

Attack methods. We follow the recent study (Balunović et al., 2021) to evaluate defences against two optimization-based attacks in FL: the classical Deep Leakage from Gradients *DLG attack* (Zhu et al., 2019), and the subsequent improved version called *GS attack* (Geiping et al., 2020) that introduces image prior and uses cosine similarity as a distance metric to enhance reconstruction. We also include one model modification attack: the recently proposed *Imprint attack* (Fowl et al., 2021).

Defense baselines. Following the recent work (Sun et al., 2021), we compare our approach with defences including traditional defences *DP* (Differential Privacy) (McMahan et al., 2017b) and *Prune* (Gradient Compression) (Lin et al., 2017). We further include the recently proposed defence *Soteria* (Sun et al., 2021) in our comparison. In Appendix A, we also provide a comparison with

Table 1: Defences against model inversion attacks on MNIST and CIFAR10. PSNR and SSIM are only computed between the sensitive data and their reconstructions. Values are averaged.

	Defense	DLG		GS		Imprint		FL
		PSNR↓	SSIM↓	PSNR↓	SSIM↓	PSNR↓	SSIM↓	Non-IID↑
MNIST	-	31.95	0.75	45.25	0.92	82.43	1.00	68.03
	Prune	9.56	0.24	11.69	0.28	75.41	0.96	66.26
	Gaussian	19.51	0.44	22.83	0.51	53.87	0.99	67.99
	Laplacian	16.17	0.38	19.45	0.44	51.35	0.99	67.80
	Soteria	11.69	0.29	11.48	0.22	82.43	1.00	68.05
	DCS ² (Ours)	6.64	0.15	10.35	0.23	19.95	0.60	67.98
	DCS ² + (Ours)	5.98	0.12	10.58	0.22	19.95	0.60	72.46
CIFAR10	-	15.85	0.52	18.50	0.65	140.62	0.98	42.02
	Prune	13.37	0.35	10.67	0.22	15.44	0.54	38.30
	Gaussian	8.14	0.12	8.73	0.14	33.21	0.86	37.83
	Laplacian	7.57	0.10	7.97	0.12	32.18	0.85	39.70
	Soteria	12.22	0.33	9.08	0.14	140.62	0.98	38.04
	DCS ² (Ours)	6.15	0.10	6.34	0.12	13.46	0.47	37.92
	DCS ² + (Ours)	5.49	0.08	6.39	0.11	14.14	0.49	41.23

defences that alter the sensitive data or modify the model’s architectures *e.g.*, PRECODE (Scheliga et al., 2022) and ATS (Gao et al., 2021). We refer to our proposed defence with only the gradient obfuscation stage as DCS², while DCS²+ includes both the gradient obfuscation stage and the gradient projection stage. Pseudocode can be seen in the Appendix B.

Datasets. We consider four datasets with different resolutions, namely MNIST (LeCun et al., 1998), CIFAR10 (Krizhevsky et al., 2009), a skin lesions dataset HAM10000 (Tschandl et al., 2018) and CelebFaces Attributes (CelebA) Dataset (Liu et al., 2015).

Models. Being consistent with the existing literature, we consider three model architectures *i.e.*, LeNet (LeCun et al., 1998) for the MNIST, ConvNet for CIFAR10 (with the same structure as in Soteria (Sun et al., 2021)) and ResNet18 (He et al., 2016) for HAM10000 and CelebA.

Metrics. To quantify the quality of reconstructed images and compare them with the original sensitive data, we use peak signal-to-noise ratio (PSNR) as used in (Balunović et al., 2021) and structural similarity index measure (SSIM) (Wang et al., 2004). Besides, we use learned perceptual image patch similarity (LPIPS) metric (Zhang et al., 2018) for experiments on HAM10000 and CelebA. 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 to measure the FL performance.

4.2 PRIVACY-PERFORMANCE TRADE-OFF

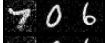



The optimal conditions for an adversary to invert gradients are a batch size of one, a low image resolution, and an untrained target network. Because our defence method needs to insert at least one concealed sample for each sensitive data, we consider the batch size to be 2 (DLG and GS attacks) and 4 (Imprint attack) with one sensitive data per batch. Our defence, therefore, learns one concealed sample in the mini-batch. We randomly select images (*e.g.*, on MNIST, we set 100 clients and randomly assign the train set to each client, then select one client to test) for evaluating the defences against attacks. We apply FedAvg (McMahan et al., 2017a) to report our results (one local update step here, but our method could be used in multiple local updates as well).

Results on MNIST and CIFAR10. Results in Table 1 indicate that, compared with existing defences, our proposed approach provides a better defence against the model inversion attacks. Specifically, on MNIST, against the DLG attack, the defence baselines reduces the PSNR from 31.95 to ~ 10 , while our defence can reduce the PSNR to around 6. When defending against the Imprint attack, the defence Soteria cannot know where the adversary would insert the imprint module, so it cannot withstand the Imprint attack. Also, our method reduces the SSIM to 0.6 when other defences only reduce it to around 0.96. In terms of the FL performance, each client only has samples with two labels. When most defences cause a performance drop, our defence retains and even enhances

Table 2: Defences against model inversion attacks on HAM10000 and CelebA. PSNR, SSIM and LPIPS are only computed between the sensitive data and their reconstructions. Values are averaged.

Defense	HAM10000				CelebA			
	Imprint			FL	Imprint			FL
	PSNR↓	SSIM↓	LPIPS↑	Non-IID↑	PSNR↓	SSIM↓	LPIPS↑	Non-IID↑
-	87.87	0.90	0.100	78.06	143.56	1.00	0.000	93.61
Prune	48.86	0.86	0.209	75.97	49.35	0.76	0.275	94.00
Gaussian	53.56	0.97	0.034	72.08	50.38	0.84	0.175	92.98
Laplacian	51.73	0.97	0.034	73.44	47.88	0.82	0.172	93.58
DCS ² +(Ours)	20.26	0.83	0.212	75.88	13.72	0.56	0.402	94.24

Table 3: Combination of our defence with DP defence against the GS attack. PSNR and SSIM are computed between all data and their reconstructions. Values are averaged.

Defense	GS			FL	
	PSNR↓	SSIM↓	Example of protected data	IID↑	Non-IID↑
-	25.16	0.64		92.08	68.03
Gaussian ($\sigma = 1e-3$)	24.69	0.63		92.07	68.00
Gaussian ($\sigma = 1e-2$)	13.64	0.31		91.91	67.99
DCS ² +Gaussian ($\sigma = 1e-3$)	12.74	0.26		93.12	72.47

it in some cases, thanks to our matching gradient projections. Specifically, on CIFAR10, when the defence baseline drops the performance by about 4%, our defence largely retains the performance and only shows a drop by less than 1% on the non-IID setting. Since DCS²+ works better on MNIST and CIFAR10, we only report the result of DCS²+ for the following experiments in the main text.

Results on HAM10000 and CelebA. We also compare different defences for higher image resolution of 224×224 , with larger capacity networks, on HAM10000 and CelebA. Because the defence Soteria cannot withstand the Imprint attack, we do not consider her here. We use randomly initialized weights for ResNet18 on HAM10000, and use the attribute gender as the target label in CelebA to perform binary classification with a pre-trained ResNet18. As shown in Table 2, our defence provides the best protection while competitively maintaining the original FL performance.

From our extensive empirical evaluations, we can conclude that our approach provides promising protection against model inversion attacks without sacrificing the performance of the Federated Learning system. Our approach is therefore promising and provides strong evidence that by protecting the privacy of the selected sensitive data, we do not necessarily need to rely on traditional defences, such as differential privacy, that cause a significant drop in the FL performance. Next, we investigate our approach in scenarios where we are required to protect the privacy of all data.

Combination for protecting all data. Here, we combine our defence with existing defences to protect all data while paying special attention to sensitive data. Taking the defence differential privacy which adds Gaussian noise into the gradients as an example, the batch size is 4 and 3 images need protection (one image is our concealed sample, *ie.* the digit ‘4’ image for Table 3). Our empirical results in Table 3, show that the combination method that crafts one concealed sample for ‘6’ and then adds Gaussian noise on the final gradients to protect all data can provide the best level of security for all data while simultaneously maintaining the FL performance.

4.3 PROPORTION OF SENSITIVE DATA

We further show the FL performance with defences against the Imprint attack by varying the proportions of sensitive data. In Figure 2, the solid line represents defence which could protect all data, while the diamond point indicates our defence method which just protects the sensitive data. All defence methods are evaluated under same setting including the number of training data, which means our defence method injects one concealed sample for one sensitive data, but the batch size is the same as other defences. As shown in Figure 2, while protecting all samples, our method

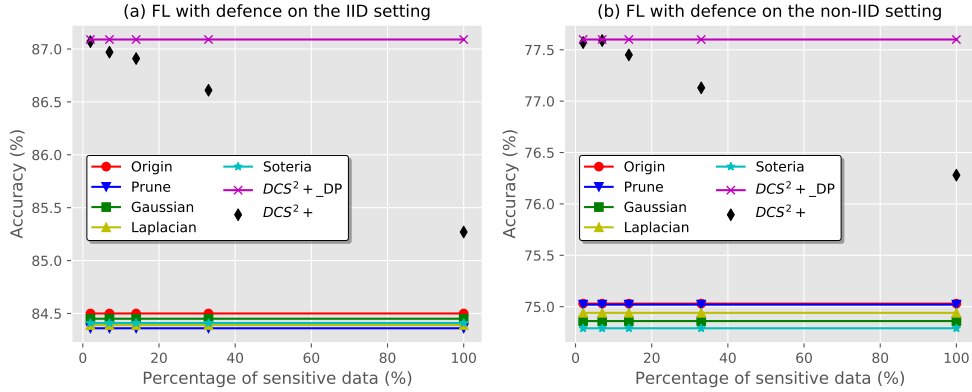


Figure 2: FL performance with defences under the different proportion of sensitive data.

Table 4: Different start points for crafting the concealed samples.

Start point	DLG		GS		Imprint		FL	
	PSNR↓	SSIM↓	PSNR↓	SSIM↓	PSNR↓	SSIM↓	IID↑	Non-IID↑
MNIST	6.00	0.12	10.58	0.22	19.95	0.60	93.11	72.52
QMNI	6.04	0.13	10.86	0.23	14.68	0.57	93.05	72.50
Noise	5.66	0.12	10.03	0.19	18.68	0.35	93.13	72.43

retains and even enhances the FL performance. Our method does need extra time and space, *e.g.*, constructing one concealed sample on MNIST costs about 8 seconds against the GS attack. So if the proportion of sensitive data is large, it is better to apply the combination defence to get the desirable privacy-performance trade-off, *e.g.*, the combination defence $DCS^2 + \text{DP}$ which only construct one concealed sample for one data but could enhance privacy for all data (as shown in Table 3).

4.4 START POINT OF CONCEALED DATA

We further evaluate our defence, by choosing different initial starting points to craft the concealed samples. As shown in Table 4, even starting from random noise, our defence method could still provide protection and retain the model’s performance. We observe that if the start point has a different distribution than the sensitive data, the proposed defence technique could be more effective (See in Appendix A), as our final solution converges to concealed samples whose gradients once inverted (by the attack) result in visually very dissimilar reconstructions.

5 LIMITATIONS

While our empirical evaluations show that our proposed defence is effective in enhancing privacy and retaining FL performance, it requires additional computation to generate concealed samples. Besides, it is unclear how to employ our algorithm in domains like natural language processing as we synthesize samples. Future directions to improve concealed samples-based defence include finding the best starting points and reducing the time to compute the concealed samples. We hope our defence can provide a new perspective for defending against model inversion attacks in FL.

6 CONCLUSION

In this work, we proposed a practical and effective defence algorithm against model inversion attacks in federated learning. 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 defences, our approach offers the best protection against model inversion attacks while simultaneously retaining or even improving the FL performance.

REFERENCES

- Mislav Balunović, Dimitar I Dimitrov, Robin Staab, and Martin Vechev. Bayesian framework for gradient leakage. *arXiv preprint arXiv:2111.04706*, 2021.
- Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchu Qiu, Titouan Parcollet, and Nicholas D Lane. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.
- Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. When the curious abandon honesty: Federated learning is not private. *arXiv preprint arXiv:2112.02918*, 2021.
- Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.
- Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoudy, Shuang Song, Abhradeep Thakurta, and Florian Tramer. Is private learning possible with instance encoding? *arXiv preprint arXiv:2011.05315*, 2020.
- Lixin Fan, Kam Woh Ng, Ce Ju, Tianyu Zhang, Chang Liu, Chee Seng Chan, and Qiang Yang. Rethinking privacy preserving deep learning: How to evaluate and thwart privacy attacks. In *Federated Learning*, pp. 32–50. Springer, 2020.
- Liam Fowl, Jonas Geiping, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models. *arXiv preprint arXiv:2110.13057*, 2021.
- Wei Gao, Shangwei Guo, Tianwei Zhang, Han Qiu, Yonggang Wen, and Yang Liu. Privacy-preserving collaborative learning with automatic transformation search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 114–123, 2021.
- Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients—how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.
- Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.
- Oded Goldreich. *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. In *International Conference on Machine Learning*, pp. 4507–4518. PMLR, 2020.
- Jinwoo Jeon, Kangwook Lee, Sewoong Oh, Jungseul Ok, et al. Gradient inversion with generative image prior. *Advances in Neural Information Processing Systems*, 34:29898–29908, 2021.
- Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. Catastrophic data leakage in vertical federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.
- Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Hongkyu Lee, Jeehyeong Kim, Seyoung Ahn, Rasheed Hussain, Sunghyun Cho, and Junggab Son. Digestive neural networks: A novel defense strategy against inference attacks in federated learning. *computers & security*, 109:102378, 2021.
- Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10132–10142, 2022.
- Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017.
- Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.
- Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017a.
- H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017b.
- Fan Mo, Anastasia Borovykh, Mohammad Malekzadeh, Hamed Haddadi, and Soteris Demetriou. Quantifying information leakage from gradients. *CoRR*, abs/2105.13929, 2021.
- Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pp. 19–38. IEEE, 2017.
- Daniel Scheliga, Patrick Mäder, and Marco Seeland. Precode-a generic model extension to prevent deep gradient leakage. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1849–1858, 2022.
- Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9311–9319, 2021.
- Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.
- Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating client privacy leakages in federated learning. In *European Symposium on Research in Computer Security*, pp. 545–566. Springer, 2020.
- Wenqi Wei, Ling Liu, Yanzhao Wut, Gong Su, and Arun Iyengar. Gradient-leakage resilient federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pp. 797–807. IEEE, 2021.
- Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo 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.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

Junyi Zhu and Matthew Blaschko. R-gap: Recursive gradient attack on privacy. *arXiv preprint arXiv:2010.07733*, 2020.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32, 2019.

A ADDITIONAL RESULTS

A.1 PROPORTION OF SENSITIVE DATA

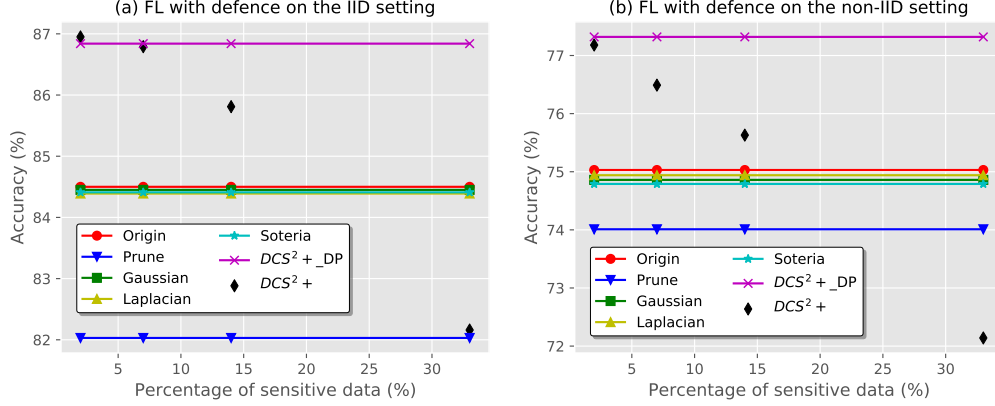


Figure 3: FL performance with defenses under the different proportion of sensitive data.

Figure 3 illustrates the performance of FL with defences against DLG and GS attacks for varying proportions of the sensitive data. The attack and the defence settings are identical to those in Table 1. When the proportion of sensitive data is less than 30%, our method can even improve the performance of FL compared to other defences. Combining our defence with differential privacy and adding Gaussian noise offers the best protection for all data points, and shows the highest FL performance.

A.2 COMPARISON WITH OTHER DEFENCES AND ATTACKS

Table 5: Performance on CelebA gender classification.

Defense	GGL		Imprint		FL
	PSNR↓	SSIM↓	PSNR↓	SSIM↓	Non-IID↑
-	11.59	0.27	155.79	1.00	74.69
Prune	10.56	0.24	140.86	1.00	78.36
Gaussian	10.56	0.24	38.64	0.99	78.71
Laplacian	10.69	0.22	36.87	0.98	76.68
Soteria	11.50	0.27	155.79	1.00	77.95
$DCS^2 +$ (ours)	8.27	0.16	19.13	0.72	79.85

Table 6: Compared with PRECODE and ATS on CIFAR10.

Defence	GS		Imprint		FL	
	PSNR↓	SSIM↓	PSNR↓	SSIM↓	IID↑	Non-IID↑
PRECODE	5.06	0.07	120.02	0.85	67.06	28.44
ATS	17.50	0.47	49.97	0.48	60.46	37.89
$DCS^2 +$ (ours)	6.43	0.11	14.14	0.49	68.05	41.86

We follow the settings in (Li et al., 2022) to compare with Generative Gradient Leakage (GGL) attack (Li et al., 2022) on CelebA, and the setting in (Balunović et al., 2021) to compare with PRECODE (Scheliga et al., 2022) and ATS (Gao et al., 2021). Results shown in Table 5 and Table 6 suggest that our defence provides the best protection with minimal drop in FL performance.

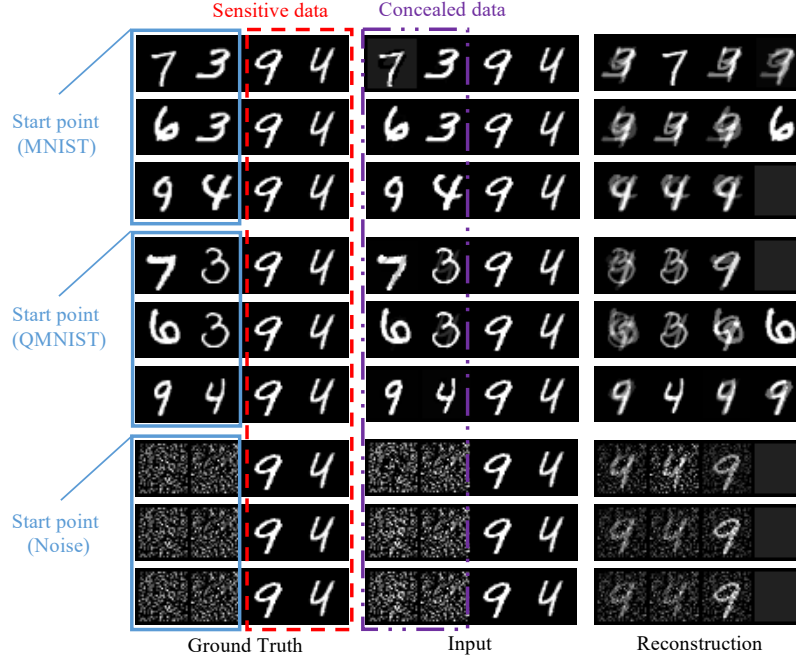


Figure 4: Examples of reconstructions for defending against the Imprint attack on MNIST when using different start points to craft the concealed data. Images in the red dashed box and purple dashed box are the sensitive data and the concealed data, respectively.

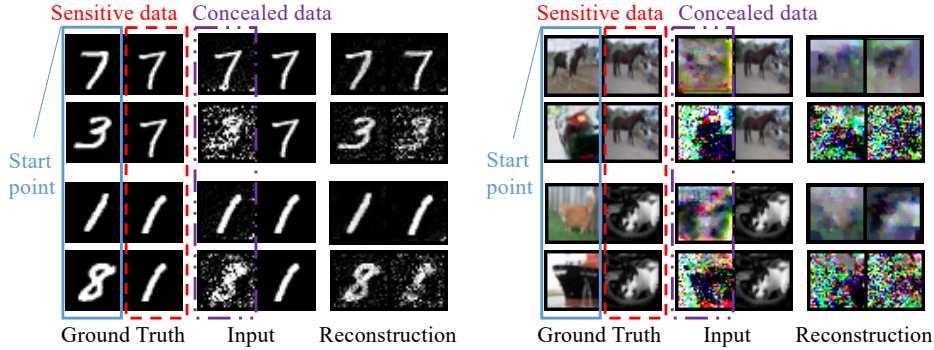


Figure 5: Examples of the reconstructions for defending against the GS attack when concealed data are computed from data with the same or different labels as the sensitive data. Images in the red dashed box and purple dashed box are the sensitive data and the concealed data, respectively.

A.3 EXAMPLES OF DIFFERENT START POINTS

Figure 4 and Figure 5 show examples of the reconstructions for defending against the Imprint attack and GS attack with different start points to craft the concealed data. To obfuscate the gradients from the sensitive data with the concealed data, generating the concealed data with starting point data sampled from different distributions is better than from the identical distributions as the sensitive data. If the start data points for computing the concealed data are sampled from the identical distributions as the sensitive data, they are likely to lie on the same side of the decision boundary and have the same gradient directions. Modifying these data points to approach the sensitive data while being visually dissimilar, therefore, becomes a challenge. As shown in Figure 5, our method cannot withstand the GS attack when the concealed data is modified from the data having the same label as the sensitive data. On the contrary, when the concealed data is crafted from the data having different labels from the sensitive data, our method can effectively defend against attacks.

A.4 ABLATION ANALYSIS ON CONCEALED SAMPLES

Here we vary the number of concealed data points.

Number of concealed data In Table 7, we report the defence results against the GS attack with different number of concealed samples for each sensitive data. The FL performance in the Non-IID setting without defences is about 68.03%, as shown in Table 7, when our defence method uses more concealed data points to imitate the sensitive data, the performance of defence against attacks is better while maintaining the FL performance.

Table 7: Defence against the GS attack with different number of concealed samples on MNIST. k denotes the number of concealed data for each sensitive data and m denotes the number of sensitive data in a mini-batch.

	$k = 1$			$k = 2$			$k = 4$		
	PSNR↓	SSIM↓	FL↑	PSNR↓	SSIM↓	FL↑	PSNR↓	SSIM↓	FL↑
$m = 1$	10.53	0.12	72.46	10.05	0.11	72.40	9.24	0.10	72.46
$m = 2$	10.40	0.09	72.40	10.21	0.09	72.40	9.75	0.09	72.26
$m = 4$	10.31	0.10	72.42	9.99	0.09	72.21	9.15	0.08	71.94

Compute Overhead In Table 8, we report the additional memory required to generate one concealed data in Table 1 and Table 2 using one GeForce RTX 3090 GPU.

Table 8: Overhead for crafting one concealed data against the attacks.

	GS attack		Imprint attack
	MNIST ($28 \times 28 \times 1$)	CIFAR10($32 \times 32 \times 3$)	CelebA($224 \times 224 \times 3$)
Time (s)	+7.7	+25.8	+8.8
Memory (MB)	+50	+90	+1072

A.5 EXAMPLES OF RECONSTRUCTIONS

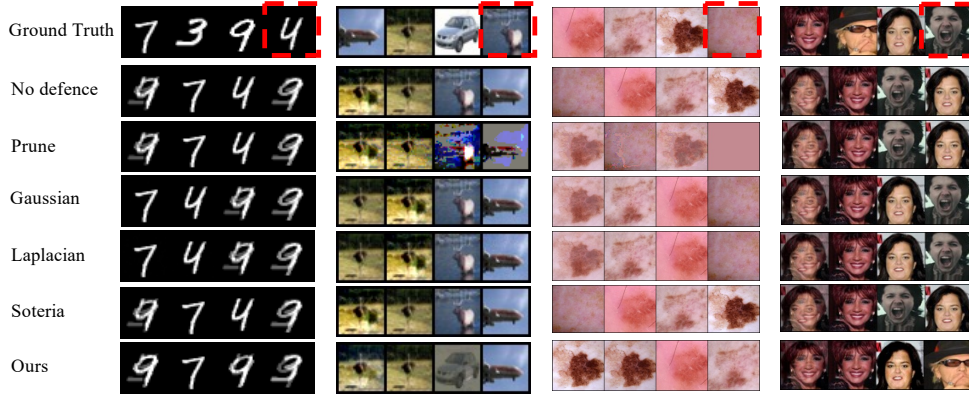


Figure 6: Examples of reconstructions for defending against the Imprint attack on MNIST, CIFAR10, HAM10000 and CelebA, respectively. The parameters for defenses are the same as those in Table 1 and Table 2. Images in red dashed box are sensitive data.

B FURTHER IMPLEMENTATION DETAILS

B.1 PSEUDOCODE OF THE PROPOSED DEFENCE METHOD

Pseudocode 1 Defence by Concealing Sensitive Samples (DCS² and DCS²+)

```

1: procedure GRADIENT OBFUSCATION
2:   initialize the start point for constructing the concealed data  $\mathbf{x}_c \leftarrow \mathbf{x}_0, \mathbf{y}_c \leftarrow \mathbf{y}_0$ ;
3:   get the concealed sample  $\mathbf{x}_c \leftarrow \text{Eq. (6)}$ ;
4:   compute the new gradient  $\mathbf{g}_c \leftarrow \text{Eq. (7)}$ ;
5: procedure GRADIENT PROJECTION
6:   get the gradient from the original batch
      $\mathbf{g} \leftarrow \nabla_{\theta} \mathcal{L}(f_{\theta}(\mathbf{X}), \mathbf{Y})$ ;
7:   if  $\langle \mathbf{g}, \mathbf{g}_c \rangle < 0$  then
8:     get the solution  $\mathbf{v}^* \leftarrow \text{Eq. (9)}$ ;
9:     project the new gradient to the closest gradient  $\hat{\mathbf{g}}_c = \mathbf{g}\mathbf{v}^* + \mathbf{g}_c$ .

```

B.2 MODEL ARCHITECTURES

Table 9: Model architectures for different datasets.

MNIST	CIFAR10	HAM10000 / CelebA
5×5 Conv, 12	5×5 Conv, 32	7×7 Conv, 64
5×5 Conv, 12	$\{5 \times 5 \text{ Conv}, 64\} \times 2$	3×3 MaxPool
5×5 Conv, 12	$\{5 \times 5 \text{ Conv}, 128\} \times 3$	$\left\{ \begin{array}{l} 3 \times 3 \text{ Conv}, 64 \\ 3 \times 3 \text{ Conv}, 64 \end{array} \right\} \times 2$
5×5 Conv, 12	3×3 MaxPool	$\left\{ \begin{array}{l} 3 \times 3 \text{ Conv}, 128 \\ 3 \times 3 \text{ Conv}, 128 \end{array} \right\} \times 2$
FC-10	$\{5 \times 5 \text{ Conv}, 128\} \times 3$	$\left\{ \begin{array}{l} 3 \times 3 \text{ Conv}, 256 \\ 3 \times 3 \text{ Conv}, 256 \end{array} \right\} \times 2$
	3×3 MaxPool	$\left\{ \begin{array}{l} 3 \times 3 \text{ Conv}, 512 \\ 3 \times 3 \text{ Conv}, 512 \end{array} \right\} \times 2$
	FC-10	7×7 AveragePool
		FC-7 (HAM10000) / FC-2 (CelebA)

Details of the models used in this study are shown in Table 9. The activation layers of the model for MNIST are Sigmoid, and for CIFAR10 and HAM10000, CelebA are ReLU.

B.3 PARAMETERS

Table 10: Parameters of different defenses against model inversion attacks.

Defense	MNIST		CIFAR10		HAM/CelebA	CelebA (32)
	DLG/GS	Imprint	DLG/GS	Imprint	Imprint	GGL
Prune	$p = 70\%$	$p = 30\%$	$p = 70\%$	$p = 70\%$	$p = 50\%$	$p = 20\%$
Gaussian	$\sigma = 1e-2$	$\sigma = 1e-2$	$\sigma = 1e-3$	$\sigma = 1e-3$	$\sigma = 1e-1$	$\sigma = 1e-1$
Laplacian	$\sigma = 1e-2$	$\sigma = 1e-2$	$\sigma = 1e-3$	$\sigma = 1e-3$	$\sigma = 1e-1$	$\sigma = 1e-1$
Soteria	$p = 5\%$	$p = 1\%$	$p = 90\%$	$p = 90\%$	-	$p = 10\%$

Attacks and defenses We build on the repository using the official implementation of the DLG, GS, GGL and the Imprint attack methods. For Soteria, Prune and DP, we build on the repository from the study (Sun et al., 2021). For ATS and PRECODE, we build upon the repository from the study (Balunović et al., 2021). Details of parameters can be found in Table 10. We set the mean and the variance of the noise distribution from the defense DP as 0 and σ , respectively. We set the pruning rate of the models’ gradients from the defense Prune and the defense Soteria as p . For our

defense method, we set $\lambda = 0.3, \alpha = 0.1, \beta = 0.001, T = 1000$ when defending against the DLG or GS attack, and $\lambda = 0.3, \alpha = 30.0, \beta = 100.0, T = 100$ when defending against the Imprint attack, $\lambda = 0.3, \alpha = 1.0, \beta = 10.0, T = 1000$ for the GGL attack.

Federated learning We build the Federated Learning (FL) framework based on the Flower (Beutel et al., 2020) platform and the FedAvg (McMahan et al., 2017a) algorithm. The details of the federated learning are shown in Table 11. For the Independent and Identically Distributed (IID) setting, the server randomly selects five from 10 clients in each round. Each client has 2000 samples for MNIST and CIFAR10, 200 samples for HAM10000 randomly sampled from the train set. For the Not Independent and Identically Distributed (Non-IID) setting, the server updates the model using gradients from the ten clients. Each client only has 400 samples for MNIST, and 4000 samples for CIFAR10 with two labels. Each label has 200 samples for MNIST, and 2000 samples for CIFAR10. For HAM10000, each client has 214, 958, 958, 594, 214, 958, 258, 214, 214, 958 samples, respectively in the Non-IID setting. For CelebA, from client 1 to client 10, each one has 170, 190, 109, 210, 151, 174, 209, 194, 235, 193 samples, respectively in the Non-IID setting. And each client has samples from 10 identities, each identity has about 0~30 images. The performance is evaluated on 10,000, 10,000, 1103 and 19962 test samples for MNIST, CIFAR10, HAM10000 and CelebA, respectively. The optimizer is SGD, and the batch size is 256 for MNIST and CIFAR10, 32 for HAM10000 and CelebA for each client, and the maximum number of training rounds is 100. Experiments about the different proportion of sensitive data are evaluated with 128 images for each client, the batch size is 32 and the FL train for 300 rounds.

Table 11: Details of the federated learning on different datasets. $|C|$, $|\tilde{C}|$, $|D_c|$, $|y_c|$ and $|B_c|$ denote the total number of clients, the number of clients selected in each round, the number of training data, the number of labels (identities for CelebA) and the batch size in each client, respectively. η and T denote the learning rate and the number of training rounds, respectively.

	Dataset	$ C $	$ \tilde{C} $	$ D_c $	$ y_c $	$ B_c $	η	T
IID	MNIST	10	5	2,000	10	256	0.01	100
	CIFAR10	10	5	2,000	10	256	0.01	100
Non-IID	MNIST	10	10	400	2	256	0.01	100
	CIFAR10	10	10	4,000	2	256	0.001	100
	HAM10000	10	10	214*958	2	32	0.001	100
	CelebA	10	10	109*235	10	32	0.001	100

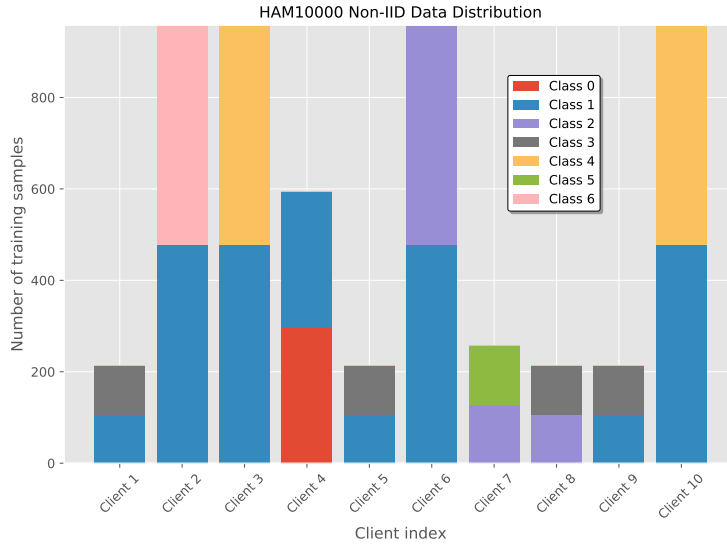


Figure 7: Non-IID data distribution on HAM10000.