CENSOR: Defense Against Gradient Inversion via Orthogonal Subspace Bayesian Sampling

Kaiyuan Zhang, Siyuan Cheng, Guangyu Shen, Bruno Ribeiro,
Shengwei An, Pin-Yu Chen[†], Xiangyu Zhang, Ninghui Li
Purdue University, [†]IBM Research
{zhan4057, cheng535, shen447, ribeirob, an93, xyzhang, ninghui}@cs.purdue.edu, [†]pin-yu.chen@ibm.com

***Abstract*—Federated learning collaboratively trains a neural network on a global server, where each local client receives the current global model weights and sends back parameter updates (gradients) based on its local private data. The process of sending these model updates may leak client's private data information. Existing gradient inversion attacks can exploit this vulnerability to recover private training instances from a client's gradient vectors. Recently, researchers have proposed advanced gradient inversion techniques that existing defenses struggle to handle effectively. In this work, we present a novel defense tailored for large neural network models. Our defense capitalizes on the high dimensionality of the model parameters to perturb gradients within a *subspace orthogonal* to the original gradient. By leveraging cold posteriors over orthogonal subspaces, our defense implements a refined gradient update mechanism. This enables the selection of an optimal gradient that not only safeguards against gradient inversion attacks but also maintains model utility. We conduct comprehensive experiments across three different datasets and evaluate our defense against various state-of-the-art attacks and defenses. Code is available at https://censor-gradient.github.io.**

I. Introduction

Federated learning (FL) gains its popularity as a privacy-preserving framework with many applications, such as next word prediction [1], credit prediction [2], and IoT device aggregation [3], etc. In FL [1], a global server broadcasts a global model to selected clients and collects model updates without directly accessing raw data. On the client side, the model is locally optimized with decentralized private training data. Once the model updates are transmitted back to the server, an updated global model is constructed by aggregating individual received models. During the whole iterative training process, raw data will not be exchanged.

Even though clients do not send their private training data in FL, the gradient updates they send are computed based on these private training data and can leak information in them. Recent studies have shown that private client data can be reconstructed through *gradient inversion* [4], [5], [6], [7], [8], [9], [10], [11]. Inversion of image from gradient information was first discussed in [12], which proved that recovery is possible from a single neuron or a linear layer. In [11], Zhu et al. showed that accurate pixel-level inversion is practical for a maximum batch size of eight. The attacker can either be an honest-but-curious server or an adversary who eavesdrops the communication channel between the server and the client to invert the private data of clients. Our paper seeks to assess the attack risks and proposes methods to mitigate privacy leakage.

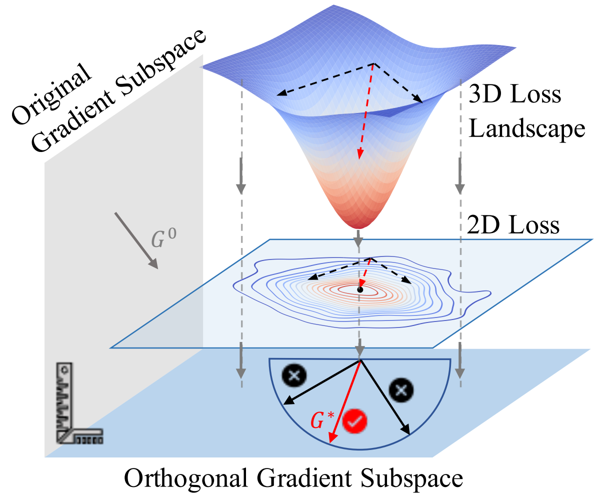

Figure 1: Intuition of cold Bayesian posteriors sampling over orthogonal subspace. The left gray plane denotes the subspace that resides original gradients G^0. The bottom blue plane signifies a gradient subspace orthogonal to the original gradient subspace (gray plane), denoted as G^*. In the middle of the figure, a 3D training loss landscape is projected onto the 2D plane, parallel to the orthogonal gradient subspace. Cold Bayesian posteriors sampling enables CENSOR to sample multiple gradients and select the optimal one. This is illustrated by three potential directions in the orthogonal subspace, with the red indicating the optimal gradient that strictly points towards the optimal loss reduction direction. Notably, the optimal gradient highlighted in red not only lies within the orthogonal subspace, effectively protecting data privacy, but also minimizes the training loss, thereby ensuring the model utility.

To mitigate the gradient inversion, researchers have proposed several defenses, including perturbing gradients with noise [13],

Network and Distributed System Security (NDSS) Symposium 2025
24-28 February 2025, San Diego, CA, USA
ISBN 979-8-9894372-8-3
https://dx.doi.org/10.14722/ndss.2025.230915
www.ndss-symposium.org

gradient clipping [14], compressing gradients [15], and using specialized gradient representations [16]. Although these techniques have demonstrated some degree of effectiveness against existing inversion attacks, it is likely that they will fail against attackers who either exploit more effective inversion attacks [4], [6], [8] or have additional prior knowledge about the private data [5], [9]. For example, if the attacker already suspects that the private instance is from a dataset known to the attacker, he can compute the gradient updates for each instance in the set and then identify which one matches the observed gradient.

In this paper, we propose a novel defense technique, CENSOR (**C**old post**E**riors co**N**trolled **S**ampling over **OR**thogonal subspace), to address the dual challenge of preserving privacy while maintaining the utility of machine learning models through a novel gradient refinement methodology. We refine model parameters by selectively updating them along an *orthogonal subspace* [17], [18] with *cold Bayesian posteriors* [19], [20]. The key insight is that Bayesian posterior sampling, which does not rely on gradient-based optimization, better resists gradient-based inversion attacks of a client's private data. And sampling inside an orthogonal subspace to the model gradients over the private data, ensures that the resulted gradients do not resemble the original sensitive ones.

Bayesian posterior sampling aims to sample the model parameters from the posterior distribution $P(\theta|D)$, which represents the probability of the parameters θ given the data D. This allows a client to obtain model parameters without resorting to gradients. However, sampling from $P(\theta|D)$ can lead to parameter samples that fit the data poorly. Cold Bayesian posteriors modify the distribution $P(\theta|D)$ by introducing a *temperature* parameter M. By tuning M, typically $0 < M \ll 1$, where $M \approx 0$ is the coldest temperature, the distribution becomes sharply peaked around the parameters that significantly reduce the loss, akin to concentrating the distribution around a Maximum a Posteriori (MAP) estimate, but retaining the ability to sample around the MAP solution.

An *orthogonal subspace* is a vector space where each vector is perpendicular to the vectors of another subspace. In our scenario, we define orthogonal subspace as the one that is perpendicular to the client's original gradient subspace. Restricting our parameter sampling to an orthogonal subspace means we will not inadvertently return the original gradients (unlike some methods as described in [13]). More precisely, in a model with m neuron weights, this orthogonal subspace can be defined by a set of $m - k$ linearly independent vectors orthogonal to the gradients of k training data instances. And, since the orthogonal subspace of a gradient of a model with m parameters has $(m - k)$-dimensions, and the resulting output of a client is a single vector in this subspace (a single gradient), the potential leakage of the original gradients is negligible if $m \gg k$.

By leveraging *cold posteriors over orthogonal subspaces*, CENSOR employs a refined gradient update mechanism. By evaluating the impact of a set of gradients inside the orthogonal subspace of the instance gradients and selecting those that sufficiently reduce the loss, our method mitigates the privacy leakage and in the mean time, keeps improving the utility of the model. Figure 1 provides an intuitive illustration of our technique.

We summarize our contributions as follows:

- We introduce CENSOR (**C**old post**E**riors co**N**trolled **S**ampling over **OR**thogonal subspace), a defense mechanism against gradient inversion attacks that operates by posterior-sampling model updates from a subspace orthogonal to the original gradients subspace. This technique does not rely on gradients and, therefore, better resists gradient reconstruction attacks, thereby preserving the privacy of the model.
- We enhance the balance between utility and privacy by incorporating *cold Bayesian posteriors* into our methodology. By adjusting the *temperature* parameter, we refine the selection process for gradient updates, narrowing the focus to those gradients that optimally maintain utility while minimizing privacy risks.
- We conduct extensive experiments to evaluate the effectiveness of CENSOR against state-of-the-art gradient inversion attacks. Our empirical results demonstrate that CENSOR not only mitigates potential privacy breaches more effectively but also outperforms state-of-the-art defenses across multiple quantitative and qualitative metrics.

Threat Model Our threat model is consistent with the literature [5], [9], [8], [6], [4], [16], [15], where the adversary is considered as an honest-but-curious server. This adversary aims to invert training samples without direct access to the data from local clients or original training data. The adversary knows the model architecture and local gradients transmitted by clients, while the adversary can not modify either the model or the gradients [21], [22]. Additionally, we assume the adversary can utilize the knowledge extracted from publicly available datasets and leverage pre-trained neural network models, such as those based on Generative Adversarial Networks (GANs) to facilitate the attack. The benign local clients can conduct defense to protect their data privacy. They have access to the model parameters at each training round but have no knowledge about the adversary's attack configuration or technique. In the end, our analysis assumes the most favorable conditions for the adversary, setting the batch size as one [8], [4]. We also adopt loose restrictions regarding the adversary, assuming they possess sufficient computational power and memory.

Roadmap. The rest of this paper is organized as follows. In Section II, we formulate the problem and introduce the background of gradient inversion attacks and defenses. In Section III, we discuss two new observations about existing gradient inversion attacks. In Section IV, we present the theoretical analysis for our technique and introduce the detailed design of our defense. In Section V, we present a comprehensive experimental evaluation of CENSOR against various attacks and compare our proposed defense with state-of-the-art defense baselines. In Section VI, we review related literature. In Section VII, we offer concluding remarks. We also provide a summary of all notations in Appendix A, Table IV for easy reference.

II. Background

In this section, we start by formulating the gradient inversion within the context of federated learning. Following this, we offer a brief overview of various existing attack techniques and several defense methods designed to counter them. Additionally, we discuss the limitations of these existing defense strategies, underscoring the necessity of introducing our approach.

A. Problem Formulation

We focus on the setting of learning a classifier using federated learning. Given a neural network f_θ parameterized by θ, the training objective is to obtain model parameters through empirical risk minimization

$$\hat{\theta} = \arg\min_{\theta} \sum_j \ell(f_\theta(X_j), Y_j), \tag{1}$$

where ℓ is the loss function, and X_j, Y_j are the input features and the corresponding label of the j-th training instance, respectively.

Within a federated learning framework, a *global server* aims to solve Equation (1) through collaborative training with multiple *local clients*, each possessing a subset of the training data. Training consists of many iterations. In the τ-th iteration, the server sends the parameter θ_τ to each client. The k-th client then computes and sends the following gradient to the server

$$G_k^\tau = \sum_j \nabla\ell(f_{\theta_\tau}(X_{j,k}), Y_{j,k}), \tag{2}$$

and the server updates its model through

$$\theta_{\tau+1} = \theta_\tau - \eta \sum_{k=1}^{N} G_k^\tau, \tag{3}$$

where N is the number of clients, and η is the learning rate. Before sending the gradient G_k^τ back to the server, local clients have the opportunity to apply defense mechanisms to it.

B. Gradient Inversion Attacks in Federated Learning.

Gradient inversion attacks present a significant threat to data privacy in federated learning systems. It enables adversaries to reconstruct private data samples of clients with high fidelity. Existing attacks can be broadly categorized into two groups: (1) Stochastic Optimization Attacks, and (2) GAN-based Attacks.

Stochastic Optimization Attacks. Attacks based on stochastic optimization invert training images from random initialization with the guidance of gradients and potential other prior knowledge. A famous work [11] conceptualized the attack vector as an iterative optimization challenge, where attackers approximate original data samples by minimizing the discrepancy between actual shared gradients and synthetic gradients derived from artificially generated data samples. Several following works have refined this attack methodology. For instance, [10] introduced a technique to infer the labels of individual data samples directly from their gradients, providing more guidance towards the ground-truth. Moreover, [5] achieved the inversion of higher-resolution images from sophisticated models like ResNet by modifying the distance metric used for optimization and incorporating a regularization term to the process.

Following existing works [5], [9], we define the gradient inversion using stochastic optimization as follows: Given a neural network with parameters f_θ, the attacker initiates the process by generating a single instance random noise X' and its label Y' as its initialization. The instance is iteratively refined to align with the ground-truth local gradients g. For simplicity, we define a function $F(\cdot)$ as deriving the gradients of the model f_θ with regards to the input, where

$$F(X') = \nabla\ell(f_\theta(X'), Y'). \tag{4}$$

There are several methods (which will be discussed later) for inferring the label Y' before the optimization process, so we exclude it from $F(\cdot)$. The stochastic optimization is driven by the goal of minimizing the discrepancy between the dummy gradients $F(X')$ and the original local gradients g that were submitted by the benign client, as described below:

$$\hat{X'} = \arg\min_{X'} \mathcal{D}\left(F(X'), g\right) \tag{5}$$

where $\mathcal{D}(\cdot,\cdot)$ is the distance metric, e.g. l_2-distance [9].

GAN-based Attacks. Recent attacks [4], [8], [23] leverage Generative Adversarial Networks (GAN) [24] to facilitate the reconstruction of high-quality images. GAN is a deep generative model, which is able to learn the probability distribution of the images from the training set. In [6], Jeon et al. proposed to search the latent space and parameter space of a generative model, effectively harnessing the generative capabilities of GANs to produce high-quality inverted images. However, it requires a specific generator to be trained for each inverted image, which may consume large amounts of GPU memory and inference time. A follow-up study [25] extends attacks on Vision Transformers. In addition, [8] adopted the generative model with label inference, which achieves semantic-level inversion. Among these GAN-based methods, only [6] addresses the scenario where the training data for the generative and global models come from different probability distributions. Intermediate Layer Optimization(ILO) [26] also proposes an optimization algorithm for solving inverse problems with deep generative models. Instead of solely optimizing the initial latent vectors, they progressively alter the input layer, resulting in increasingly expressive generators. Following ILO, Fang et.al [4] exploits pre-trained generative models as data prior to invert gradients by searching the latent space and the intermediate features of the generator successively with l_1 ball constraint, address the challenges of expression ability and generalizability of pre-trained GANs.

Following existing works [8], [6], [4], we define GAN-based attacks by assuming that the attacker has access to a pre-trained generative model, which is trained using a large public dataset. The problem can be formulated as follows:

$$z^* = \arg\min_{z\in\mathbb{R}^k} \mathcal{D}(\mathcal{T}(F(G_p(z))), g) + \phi(G_p; z), \tag{6}$$

where G_p denotes the pre-trained generative model and $z \in \mathbb{R}^k$ represents its latent space. $\mathcal{T}$ is a gradient transformation function such that the attackers can adaptively counter the effects of defense strategies. $\phi(\cdot)$ represents a regularization term that imposes penalties on latent vectors that diverge from its original prior distribution of G_p.

Label Inference Assisting Gradient Inversion. Several existing attacks, both stochastic and GAN-based, leverage label inference to enhance the fidelity of inversion. These attacks show that private labels can be directly inferred from the gradients [10], [9], [8]. Utilizing the shared gradients, the adversary initially employ analytical inference techniques [10], [8] to infer the ground-truth label c of the client's private image. Adopting notation from [8], we present the label inference process for federated learning (FL) models engaged in a classification task across C classes. The computation for the i-th entry of the gradients associated with the weights of the final fully-connected (FC) classification layer (denoted as ∇W_{FC}^i) is described by the following equation:

$$\nabla W_{\text{FC}}^i = \frac{\partial \ell(f_{\theta^\tau}(X), Y)}{\partial z_i} \times \frac{\partial z_i}{\partial W_{\text{FC}}^i} \tag{7}$$

where z_i denotes the i-th output of the FC layer. Note that the calculation of the second term, $\frac{\partial z_i}{\partial W_{\text{FC}}^i}$, yields the post-activation outputs from the preceding layer. When activation functions such as ReLU or sigmoid are employed, these outputs will always be non-negative. When networks are trained with cross-entropy loss using one-hot labels (and assuming softmax activation is applied in the final layer), the first term will only be negative when $i = c$. Consequently, the ground-truth label can be identified by locating the index where ∇W_{FC}^i is negative.

C. Defenses Against Gradient Inversion Attacks

Next, we introduce several current defense mechanisms against gradient inversion attacks.

Noisy Gradient. Differential privacy (DP) is the established method for quantifying and controlling the privacy exposure of individual clients. In federated learning, DP [13] can be applied at either the server's side or the client's side. Clients can employ a randomized mechanism to modify the gradients before they are shared with the server. DP provides a theoretical worst-case guarantee about the amount of information an adversary can extract from the data released. However, achieving these worst-case guarantees often requires substantial additive Gaussian noise to the gradient [27], [28], which may significantly compromise the utility of the global model.

Definition II.1 (Differential Privacy)**.** *A randomized mechanism $\mathcal{M} : \mathcal{A} \rightarrow \mathcal{R}$ with domain $\mathcal{A}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy if, for any two adjacent inputs $d, d' \in \mathcal{A}$ and for any subset of outputs $S \subseteq \mathcal{R}$, it holds that*

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(d') \in S] + \delta,$$

where $\epsilon > 0$ is the privacy budget and $\delta \geq 0$ is the privacy loss parameter.

Gradient Clipping. Wei et al. [14] introduces a defense mechanism resilient to gradient leakage, utilizing a client-level differential privacy (DP) approach. In each layer l, the gradient g_l is computed for every individual training example, and the clipping transformation function is defined as

$$\mathcal{T}(g_l, b) = g_l \cdot min(1, b / \|g_l\|_2), \tag{8}$$

where b is the constant representing the upper bound for clipping. This method applies per-example gradient clipping, followed by the addition of a Gaussian noise vector to the clipped gradient, to secure demonstrable differential privacy guarantees. Wei et al. have demonstrated that gradient clipping can achieve client-level per-example DP and that the approach is resistant to gradient inversion. However, existing attacks [8], [4] have empirically indicated that gradient clipping defense mechanisms are often ineffective in federated learning contexts. The reason is that the attacker can estimate the clipping bound by computing the l_2 norm of the gradients received for each layer, thereby revealing strong guidance towards the ground truth gradient.

Gradient Compression / Sparsification. In Top-k compression [15], the client only selects the k largest components of a gradient g in terms of absolute values and sets all other entries to zero. This technique defines the compressed gradient g' such that each component g_i' of g' is determined by the following:

$$g_i' = \begin{cases} g_i & \text{if } g_i \in \text{Top}_k(\{|g_1|, |g_2|, \ldots, |g_n|\}) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

This operation effectively retains only the k components with the highest magnitudes for gradient transmission. Existing attacks [11], [8], [4] have shown that gradient sparsification is not always effective. Since the implementation involves applying a gradient mask, the attackers is able to estimate the gradient's sparsity by observing the percentage of non-zero entries in the shared gradients, thereby leaking information of the ground-truth gradients.

Gradient Representation. Recent work [16] identifies gradient-induced data leakage in federated learning and introduces a defensive strategy called Soteria. This approach calculates gradients using perturbed data representations. Let X and X' represent the original and inverted images via perturbed gradients, respectively, with r and r' as their corresponding representations in the protected layer. Soteria aims to minimize information leakage by maximizing the distance between X and X' while ensuring that the representations r and r' remain similar. The optimization problem is formulated with constraints as follows:

$$\begin{aligned} \max_{r'} \quad & \|X - X'\|_2 \,, \\ \text{s.t.} \quad & \|r - r'\|_0 \leq \epsilon \end{aligned} \tag{10}$$

In Soteria, the l_0 norm ensures sparsity, effectively acting as a pruning rate in the gradient sparsification mechanism. The defense involves an optimization phase that introduces substantial computational overhead, particularly within the fully-connected layers. Despite this, Soteria maintains robustness and does not hinder the convergence of federated learning systems [16]. However, the defense mechanism is essentially equivalent to apply a gradient mask to the protected layer. Once the global model f_θ and input X are given, the masking process is deterministic. Consequently, an attacker could easily reverse engineer the original gradients in the protected layers.

III. Observations on Existing Attacks

The literature observes several effective gradient inversion attacks that underscore significant privacy threats. At the same time, researchers have identified limitations in these attacks. For example, Huang et al. [29] identified two assumptions that IG [5] and GI [9] rely upon for effectiveness. First, these attacks assume that the adversary knows the batch normalization (BN) statistics, such as mean and variance of the input instance to be constructed. Second, they assume the adversary knows or is able to precisely infer the label of input instance and use that information in reconstruction. These information is typically unavailable in practical real-world scenarios. It was shown that by nullifying these assumptions, the performance of the attacks degrades significantly, only working for low-resolution images.

In this section, we further introduce two additional observations, serving as the basis for our defense design. We first implement and evaluate five state-of-the-art gradient inversion attacks, including stochastic optimization attacks IG [5] and GI [9], and GAN-based attacks GGL [8], GIAS [6], and GIFD [4]. We aim to better understand the strengths and limitations of these gradient inversion attacks, in order to help us more effectively defend against them. We focus on the setting where the clients use a batch size of one; i.e., each client sends to the server the gradient of a single instance and the adversary tries to reconstruct this instance from the gradient. We prioritize this setting because we aim to develop defense mechanisms capable of countering attacks in this particularly challenging context, which is the easiest case to attack and thus the hardest to defend against.

In our study, we make two new observations about existing inversion attacks: ① Most existing attacks succeed only within the early stage of training; ② GGL is the only attack that is able to produce *high-quality* images beyond the first few epochs. GGL achieves this by first inferring the label of the input instance; however, it typically reconstructs images that are independent of the ground-truth input instance. Taken these two observations together, we conclude that the threat of gradient inversion attacks mostly lies during the first few training epochs and that the defense needs to be improved to protect the label leakage. Next, we elaborate on these observations.

Observation ①: Most existing attacks succeed only within the early stage of training. Most existing attacks [8], [6], [4] only evaluate the inversion performance at the first epoch 0, assuming random parameter initialization. To evaluate the robustness of these attacks in a more realistic federated learning scenario, we design an experiment with over a thousand clients. Each client processes data with a batch size of 1. To speed up the training within this simulation, we employ a setup consisting of one *victim* client with a batch size of 1, and another local client with a larger batch size of 1024. Both clients utilize a stochastic gradient descent (SGD) optimizer, which is

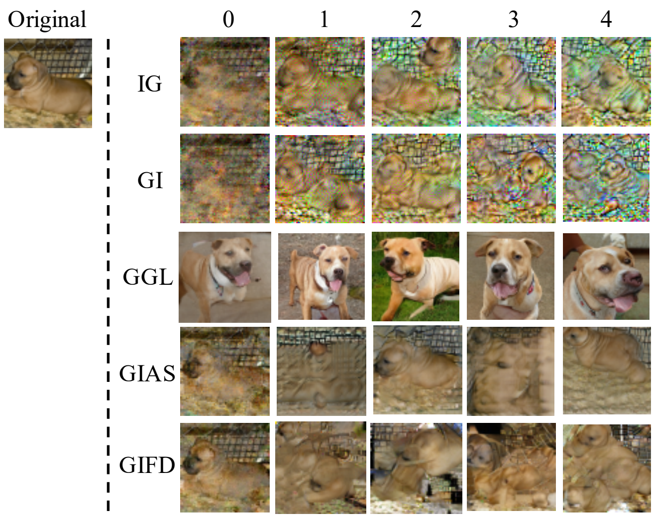

Figure 2: Results of SOTA attacks inversions (batch size equals to 1) across the initial five epochs.

commonplace in standard federated learning frameworks [1]. As the number of training rounds increase, the ability of an attacker to invert and extract useful information diminishes. This configuration suggests relative safety for benign clients when submitting gradients during the later converged stages of training. *However, the inherently non-i.i.d. nature of federated learning often results in scenarios where some clients possess only one or a few training instances.* Concern arises when such a benign client participates in the initial stages of federated learning. Besides, batch size of 1 is the easiest setting to attack and the hardest to defend against. The security of this client's raw data against inversion attacks, under these conditions, remains questionable. Given these complexities, it is imperative to evaluate the security implications more rigorously.

The results of the experiment is illustrated in Figure 2, where each row corresponds to a different attack, with columns to the right of the dotted line representing the rounds of federated learning training. Note that no defense is applied here, and we are investigating the vanilla performance of the various attacks. We show the experiment result using a batch size of 1 over 5 FL training rounds. Observe that while certain attacks successfully invert data during the initial epochs (e.g., the first two), others are confined to reconstructing only partial details (such as eyes or texture features). Notably, we find that the only attack that is able to produce high-quality images, but not faithfully same as the original instances in the first few epochs, is GGL [8], as shown in the third-row image of Figure 2. This phenomenon is further explored in the subsequent observation.

This empirical evidence underscores a critical insight:

Privacy leakage persists, even though the effectiveness of attacks decreases in the early training epochs. *Securing the initial training phase for benign local clients, especially at batch size of 1 for the easiest setting to attack, poses significant challenges.*

In light of this, our defense strategy, Censor, is strategically implemented within the first few epochs. In our evaluations, we also opt for conditions most favorable to the attacker, specifically a batch size of one, to demonstrate the robustness of our approach in protecting against gradient inversion attacks.

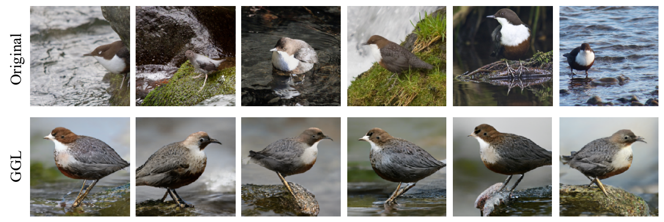

Figure 3: GGL consistently inverts similar images across distinct inputs.

Observation ②: GGL [8] inverts high-quality but typically low-fidelity images, and reveals label information. From the above analysis and the illustrations in Figure 2, the GGL [8] attack uniquely manages to produce high-quality images even after the initial epochs. While one could interpret this as GGL being an especially effective gradient inversion attack, from Figure 2 we observe that the reconstructed images, while being clear representatives of the label of the original instance, do not look particularly similar to the original instance.

We conduct additional experiments to analyze this effect. Figure 3 shows the result, where the first row shows six different training instances of the class bird, and the second row shows the corresponding instances reconstructed by GGL using gradients submitted in the first epoch 0. Note that while the second row shows high-quality bird images, these images are similar to each other, and notably distinct from the ground-truth image in the first row. This is because GGL is able to successfully infer the label of the original instance, and then use the robust generative capabilities of GANs to produce well-formed images that are good representatives of the label. On the other hand, while GGL may not faithfully reconstruct the original training samples, it does identify the ground-truth label and provides representative instances of that label. Therefore, it still to some extent compromises the privacy of the local training data.

This empirical evidence highlights a critical insight:

Privacy leakage persists, GGL inverts low-fidelity images, yet reveals label information. *The GGL attack, despite primarily producing low-fidelity images, effectively infers label information and provides label-representative instances through its strategic exploitation of the latent space using a pre-trained GAN. This capability highlights a significant challenge for defense mechanisms, which struggle to adequately prevent label leakage despite existing efforts to obfuscate or alter the data representation.*

Our proposed methodology, CENSOR, introduces a novel defensive mechanism that strategically employs an alternative gradient vector located within a high-dimensional space, specifically within a subspace orthogonal to the original gradient vector. In other words, CENSOR presents a radically different gradient profile from an orthogonal subspace, which results in seriously corrupted gradient information, rendering GGL unable to precisely infer the label information and perform any effective inversion.

Summary. Figure 2 shows that all state-of-the-art attacks other than GGL reconstructs low-quality images except for the first few epochs, and Figure 3 shows that GGL's high-quality reconstructed images are not high-fidelity reconstructions of the input instance, but rather representative instances of the label. Going forward, we thus focus on defending against gradient inversion attacks during the first few training epochs and enhance the protection against label leakage.

IV. Proposed Defense: Gradient Sampling over Orthogonal Subspaces

In this section, we first lay foundation of the theoretical analysis for our defense technique, CENSOR. Then we introduce the details of our proposed defense, which takes advantage of the high dimensionality of the model parameters to sample gradients in a *subspace* that is *orthogonal* to the original gradient, such that the new gradient also reduces the loss like the original gradient but is not in the same direction.

A. Learning over Orthogonal Subspaces

Rationale: Learning through Bayesian Updates on Orthogonal Subspaces Instead of Gradients. Statistically, the minimization described in Equation (1) that reduces the loss $\ell(\cdot)$ —cross-entropy or L2 loss— is equivalent to maximizing the log-likelihood of categorical or Gaussian distributions [30, pp. 25–27], which is also equivalent to minimizing the negative log-likelihood of the (neural network) model.

Maximizing the likelihood in neural networks uses gradients to update the model. Interestingly, in contrast to maximizing the likelihood, which seeks to find a single optimal set of neuron weights, a Bayesian approach yields a posterior distribution $P(\theta|D)$ over the neural network parameters θ, conditional on the training data D. The neural network parameter prior $P(\theta)$ is usually a standard (isotropic) Gaussian. This posterior distribution can be sampled and the resulting sampled parameters can be used for predictions.

The key insight of CENSOR is to develop a Bayesian-inspired procedure to update the global model that is resistant to gradient inversion attacks. If client $k \in \{1, \ldots, K\}$ can somehow sample from a modified posterior $P(\theta'|D^k)$ (not necessarily the true posterior $P(\theta|D^k)$) to produce updates to the global model, the client can send the global model an update that is not based on gradients. *If the update is not a gradient, then gradient inversion attacks would struggle to invert it.*

In order to avoid gradients, we first need to overcome some obstacles:

In the following exposition, the terms $P(D|\theta)$ and $P(\theta|D)$ are described as normalized probabilities. It is important to note, however, that the equations are equally applicable to probability densities and unnormalized probabilities.

Obstacle 1: Bayesian neural networks [31], [20] offer a gradient-free alternative to maximum likelihood, but they are not without drawbacks [32]. In overparameterized neural networks, even if the posterior probability $P(\theta|D)$ of sampling poorly-performing neural network parameters is small —i.e., f_θ has a large loss (equivalently a small probability $P(D|\theta'))$— in overparameterized neural networks these are still likely to be sampled due to the high dimensionality of θ.

A simple solution is noticing that the maximum a-posteriori (MAP) solution $\hat{\theta}_p = \arg\min_\theta - \log P(\theta|D) = \arg\min_\theta - \log P(D|\theta) - \log P(\theta)$ is very close to the maximum likelihood estimation (MLE) solution $\hat{\theta} = \arg\min_\theta - \log P(D|\theta)$ obtained by minimizing the loss with gradient descent, if the prior $P(\theta)$ is weak (which is our case) [20]. However, employing MAP reverts the model update back to gradient-based optimization. Thankfully, there is a middle-ground: *cold posteriors*. Cold posteriors are a hybrid between sampling parameters from the original posterior $P(\theta|D)$ and the MAP gradient-based solution. Cold posteriors make use of the fact that the modified posterior $P^M(\theta|D) \propto P(D|\theta)^M P(\theta)$ is highly concentrated around the MAP solution $\hat{\theta}_p$ when the "temperature" parameter M is close to zero but still allows Bayesian sampling [19]. By setting the temperature parameter M close to zero, we can sample neural network parameters from the Bayesian posterior without sacrificing too much the quality of the model.

Obstacle 2: **The global server expects gradients from its clients.** Here, we design an acceptance/rejection sampling procedure inspired by Metropolis-Hastings's rejection sampling that outputs parameter updates that can be used like gradients by the global server. By sampling a perturbation $G \sim \mathcal{N}(\mathbf{0}, \epsilon I)$ to the current global model parameters θ_τ, for $\epsilon \in R^+$ small, and then accepting the sample G with probability

$$P(\text{accept}) = \min\left(1, \frac{P(D^k|\theta_\tau + G)^M P(\theta_\tau + G)}{P(D^k|\theta_\tau)^M P(\theta_\tau)}\right),$$

where θ_τ is the current global model at step τ at client k and $M > 0$ is the temperature described earlier; otherwise, restart the sampling-acceptance process. The Metropolis-Hastings samples behave as if sampled from $P^M(\theta|D)$. Using rejection sampling will allows us to have some flexibility defining the "gradient" perturbation G later.

Our approach avoids rejections by sampling T proposed random perturbations $G'^1, \ldots, G'^T$ and, since we are interested in cold posteriors, client k simply selects the best proposal $G^\star = \arg\max_{G \in \{G'^1,\ldots,G'^T\}} P(D^k|\theta_\tau + G)$. This approach, however, has a potential challenge: For $T \gg 1$, $G^\star$ may be similar to the MAP gradient at θ_τ, which could be inverted by the adversary. In what follows we address this potential drawback.

Obstacle 3: **Actively avoiding samples too close to original gradient.** This is achieved by sampling perturbations $\{G^1, G^2, \ldots, G^T\}$ that are provably orthogonal to the true gradients at θ_τ: $\{\nabla \ell(f_{\theta_\tau}(X_{j,k}), Y_{j,k})\}_j$. We obtain $\{G^1, G^2, \ldots, G^T\}$ by projecting $\{G'^1, G'^2, \ldots, G'^T\}$ onto the orthogonal subspace of the original gradients of client k. More precisely, we project them into the orthogonal subspace

$$\begin{aligned} S_k^\perp = \text{span}\{\forall w \in V : \langle w, \nabla l(f(X_{j,k}, \theta)), c)\rangle = 0, \\ \forall (X_{j,k}, \cdot) \in D^k, \forall c \in [1, C]\}, \end{aligned} \tag{11}$$

where $\langle\cdot,\cdot\rangle$ is the inner product, and $S_k^\perp$ is obtained through the Gram-Schmidt algorithm. An *orthogonal subspace* can be understood as a vector space where each vector is perpendicular to the vectors in another subspace [33], [34], [17]. For example, in an m-dimensional space represented by $x_1, \ldots, x_m$, if one subspace consists of vectors along the x_1-axis, an orthogonal subspace might be confined to the $x_2, \ldots, x_m$-plane. This arrangement ensures that changes or optimizations made within one subspace (along the x_1-axis) have no direct impact on the vectors lying in the orthogonal subspace (within the $x_2, \ldots, x_m$-plane).

Complete solution: From the sampled set $\{G^1, G^2, \ldots, G^T\}$ of the orthogonal perturbations, we select the optimal $G^\star = \arg\max_{G \in \{G^1,\ldots,G^T\}} P(D^k|\theta_\tau + G)$. By integrating this selection with orthogonal gradients, our method ensures both effective loss minimization and safeguards against gradient inversion attacks, protecting the client's private data from potential breaches. The pseudocode of our complete algorithm CENSOR is shown in Algorithm 1, with more details covered in the next few paragraphs.

B. CENSOR *Details*

Consider a benign local client. She trains a model on her local dataset and achieves a gradient vector that minimizes the training loss. She then submits her gradient to the honest-but-curious global server, the global server may invert her raw training data from the submitted gradient. We address the scenario where the benign local client aims to protect the privacy of local training data without significantly reducing model accuracy. In high-level, CENSOR samples gradients in a *subspace* that is *orthogonal* to the original gradient layer by layer and select the one that achieves the lowest loss. The overview of CENSOR is illustrated in Figure 4. CENSOR conducts defense to safeguard local data privacy, layer by layer. Beginning from the left and moving towards the right of Figure 4, the local client computes the original gradient of the input image through back-propagation. CENSOR then transforms the original gradients into protected ones, as depicted in the highlighted pink area. Specifically, CENSOR generates a set of new gradients that are orthogonal to the original one. Subsequently, it undergoes a loss analysis to select the gradient that minimizes the training loss, and returns the optimal gradient. While an attacker may invert the original image based on the original gradient, it fails to produce a meaningful image for the protected gradient.

In addition to the overview, we offer a detailed description of our approach in Algorithm 1, outlined below.

FL Training Paradigm. We formally define the typical FL training paradigm in Line 1-7. Line 1 specifies the input of the global server where f denotes the global model function, η is

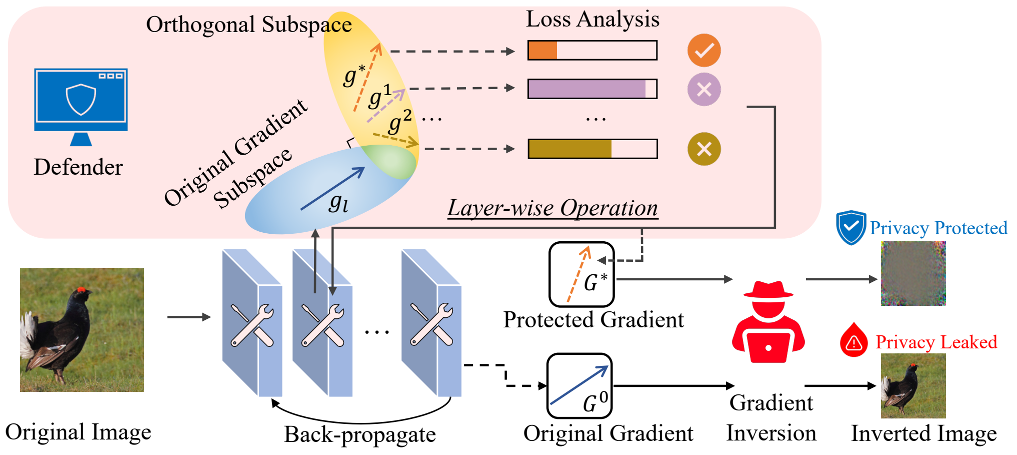

Figure 4: Overview of CENSOR.

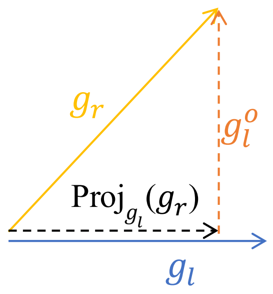

Figure 5: Orthogonal projection.

the learning rate, and R is the total number of training rounds. At each round τ, the global model will randomly select a set of K clients for training. Line 2 introduces the local client's input. It has its own local dataset D^k and optimizes the global model parameter θ_τ at round τ. In Line 3-5, each local client k derives its local update based on the local data for each round τ. Function `LOCAL_UPDATE` specifies this procedure. It then submits the local update to the global server and the global server aggregates the received gradients in Line 6. Finally, the global server update the global model parameters in Line 7.

CENSOR's Calculation of Local Updates. CENSOR mitigates the local data privacy leakage and operates on the benign local clients. Line 8-20 describe CENSOR's calculation of the local update based on the current global model parameters and the local data. It consists of 3 principle phases: (1) Obtaining layer-wise orthogonal gradient updates, (2) Normalization, and (3) Selecting the best gradient according to the loss decrease. Before delving into the main phases, CENSOR first derives the original gradient without defense in Line 9, and initializes the best loss in Line 10. Function `EVALUATE`, detailed in Line 21-23, specifies how to calculate the initial loss. In Line 11, CENSOR initializes the best gradient using the original one. Line 12-19 detail the main process. To obtain a best local update, CENSOR takes a number of trials T and selects the best gradient for submission. Usually $T = 20$ is sufficient. We perform ablation study on the number of trials in Section V-E.

Phase 1: Layer-wise Orthogonal Gradient Update. To prevent gradient inversion, CENSOR produces gradient update that is orthogonal to the original one. Line 13 denotes the procedure, where Function `ORTHOGONAL_GRAD` takes the original gradient G^0 and returns a orthogonal one, layer by layer. Details of the function are presented in Line 24-30. CENSOR initializes the gradient for the entire model in Line 25. In Line 26, CENSOR takes the gradient g_l in each layer. It then samples a new gradient g_r with the same shape of g_l from a normal distribution $\mathcal{N}$ in Line 27. To ensure the orthogonality, CENSOR projects g_r on g_l in Line 28 and append the orthogonal fraction to G^t in Line 29.

The projection is illustrated in Figure 5, where CENSOR derives the orthogonal projection by subtracting the component of g_r that aligns with g_l's direction ($\text{Proj}_{g_l}(g_r)$), and then derives the orthogonal component g_l^o. In addition, we introduce a formalization to clarify the projection process utilized in the preceding steps. Specifically, the projection of vector g_r in the direction of vector g_l is defined as $\text{proj}_{g_l}(g_r) = \frac{\langle g_r, g_l \rangle}{\langle g_l, g_l \rangle} g_l$. Applying this formalism, the orthogonal gradient for layer l, g_l^o, is computed as:

$$g_l^o = g_r - \text{proj}_{g_l}(g_r) = g_r - \frac{\langle g_r, g_l \rangle}{\langle g_l, g_l \rangle} g_l. \tag{12}$$

Phase 2: Normalization. Besides orthogonal projection, CENSOR incorporates layer-wise normalization according to the original scale, aiming to avoid some of the gradient matrix including significant large values. Line 14 presents the normalization phase, which is elaborated on in Lines 31-36. CENSOR initializes the returning gradient of the entire model in Line 32. For the orthogonal gradient g_l^o and the original one g_l in each layer, CENSOR normalizes g_l^o to the scale of g_l and derives $\tilde{g}_l^o$ in Lines 33-34. It then appends the normalized layer-wise gradient to G_N^t in Line 35.

Phase 3: Gradient Selection. To ensure a positive local update to the global model, CENSOR selects the best normalized orthogonal gradient in all trials, according to their contribution to the loss reduction. The selection process is presented in Line 15-19. In Line 15, CENSOR applies a gradient candidate G_N^t to the model and evaluate its loss in Line 16. It then compares the current loss ℓ^t with the best loss $\ell^\star$ (Line 17), and updates the best loss $\ell^\star$ (Line 18) and the best gradient $G^\star$ (Line 19) if ℓ^t is smaller than $\ell^\star$.

Finally, the optimal gradient $G^\star$ is returned (Line 20) as the protected local update. CENSOR guarantees the privacy of local data by employing orthogonal projection and maintains global model performance through normalization and loss-guided gradient selection.

Algorithm 1 Pseudocode of CENSOR

```
1: Global Server Input: Global model function $f$, the learning rate $\eta$, total FL training round $R$, and a set of randomly selected clients $\{1, 2, \cdots, K\}$ at round $\tau$
2: Local Client Input: The $k$-th client's local dataset $D^k$, where $1 \leq k \leq K$, and the global model parameters $\theta_\tau$ at round $\tau$
3: for each training round $\tau$ in $\{1, 2, \cdots, R\}$ do
4:     for each client $k$ in $\{1, 2, \cdots, K\}$ do
5:         $G^k_{\tau+1} \leftarrow$ LOCAL_UPDATE($\theta_\tau, D^k$)   ▷ $k$-th client trains on her local data and submits the update to the global server
6:     $G_{\tau+1} = \frac{1}{K}\sum_{k=1}^{K} G^k_{\tau+1}$   ▷ Global server aggregates the received gradients from local clients
7:     $\theta_{\tau+1} = \theta_\tau - \eta \cdot G_{\tau+1}$   ▷ Update of the global model parameters
8: function LOCAL_UPDATE($\theta_\tau, D^k$)
9:     $G^0 = \nabla_\theta f(D^k, \theta_\tau)$   ▷ Derive the original gradient
10:    $\ell^\star \leftarrow$ EVALUATE($f_{\theta_\tau}, D^k$)   ▷ Initialize the best loss
11:    $G^\star = G^0$   ▷ Initialize the best gradient
12:    for each trial $t$ in $T$ do ▷ $T$ is the number of trials
13:        $G^t \leftarrow$ ORTHOGONAL_GRAD($G^0$)   ▷ Phase 1
14:        $G^t_N \leftarrow$ NORMALIZE_GRAD($G^t, G^0$)   ▷ Phase 2
15:        $\theta^t = \theta_\tau - \eta \cdot G^t_N$   ▷ Phase 3
16:        $\ell^t =$ EVALUATE($f_{\theta^t}, D^k$)   ▷ Evaluate the effect of the current gradient
17:        if $\ell^t < \ell^\star$ then   ▷ Update the best loss and gradient
18:            $\ell^\star = \ell^t$
19:            $G^\star = G^t_N$
20:    return $G^\star$
21: function EVALUATE($f_\theta, D^k : \{x, y\}$)
22:    $\ell = \mathcal{L}(f_\theta(x), y)$ ▷ $\mathcal{L}$ is the loss function, e.g., Cross Entropy
23:    return $\ell$
24: function ORTHOGONAL_GRAD($G^0$)
25:    $G^t = \{\}$
26:    for $g_l$ in $G^0$ do ▷ Layer-wise orthogonal gradient projection
27:        Sample $g_r \sim \mathcal{N}$   ▷ $\mathcal{N}$ denotes a normal distribution
28:        $g^o_l = g_r - \text{proj}_{g_l}(g_r)$   ▷ Project $g_r$ onto $g_l$
29:        $G^t = G^t \cup \{g^o_l\}$   ▷ Append layer-wise gradient back to the gradient set
30:    return $G^t$
31: function NORMALIZE_GRAD($G^t, G^0$)
32:    $G^t_N = \{\}$
33:    for each pair $(g^o_l, g_l)$ in $(G^t, G^0)$ do ▷ Layer-wise operation
34:        $\tilde{g}^o_l = g^o_l \cdot \frac{\|g_l\|_2}{\|g^o_l\|_2}$   ▷ Normalize $g^o_l$
35:        $G^t_N = G^t_N \cup \{\tilde{g}^o_l\}$
36:    return $G^t_N$
```

V. EVALUATION

In this section, we provide a comprehensive empirical evaluation of our proposed defense CENSOR. We outline the experimental setup in Section V-A. Section V-B presents the assessment of CENSOR's defense effectiveness against five attacks and compares it with four state-of-the-art defenses, across different batch sizes. We perform a convergence study in *non-i.i.d.* federated learning setting in Section V-C. In Section V-D, we examine CENSOR's effectiveness against adaptive attacks. Section V-E provides several ablation studies to investigate the impact of different design components and hyper-parameters of CENSOR.

A. Experimental Setup

To validate CENSOR's defense performance, we conduct experiments using five state-of-the-art attacks: IG [5], GI [9], GGL [8], GIAS [6], and GIDF [4]. We compare CENSOR with four defense baselines: Noise [13], Clipping [14], Sparsi [15], and Soteria [16]. We following existing attacks to adopt a randomly initialized ResNet-18 [35] as the FL initial model, and employ the negative cosine similarity as distance metric $\mathcal{D}(\cdot)$ to measure the discrepancy between the inverted gradient and the original version. To ensure fairness, we strictly follow the official implementation of both the attacks and the baseline defenses. Our evaluation covers three widely-used dataset: ImageNet [36], FFHQ [37] (10-class, using age as label) and CIFAR-10 [38]. We use batch size $B = 1$ as the default setting, representing the easiest scenario for the attacker and most challenging for the defender. The default number of sampling trials for CENSOR is 20, and we conduct an ablation study on this parameter in Section V-E. Perturbed gradients are drawn from a normal distribution $\mathcal{N}$. Additionally, these gradients can also be constructed in alternative ways, such as from other meaningful images to mislead the attacker. We present the results of this experiment in Section V-E. We will release our code upon publication.

Attacks Configurations. (1) Inverting Gradients (IG) [5] uses Adam on signed gradients, with cosine similarity to optimize the input initialized from Gaussion; (2) GradInversion (GI) [9] initializes the pixels from Gaussian noise and uses Adam to optimize them with gradient matching; (3) Gradient Inversion in Alternative Spaces (GIAS) [6] uses negative cosine as the gradient dissimilarity function and the same Adam is used as the optimizer; and (4) Generative Gradient Leakage (GGL) [8] applies BigGAN [39] on ImageNet and CIFAR-10, and StyleGAN2 [37] on FFHQ. They also use KL-based regularization and CMA-ES optimizer; (5) Gradient Inversion over Feature Domains (GIFD) [4] uses intermediate features of BigGAN or StyleGAN2 and utilizes Adam optimizer with a warm-up strategy. They apply regularization with ℓ_2 distance.

Defenses Configurations. We follow the same defense setup as the previous work [4], [8], [6]: (1) Gaussian Noise [13] randomly perturbs the gradients using Gaussian noises with a standard deviation of 0.1; (2) Gradient Clipping [14] constrains the magnitude of gradients by clipping each value within a clipping bound; (3) Gradient Sparsification [15] maps small absolute gradients to zero and only transmits the largest values during update; and (4) Soteria [16] prunes by applying a mask to the defended fully connected layer's gradients.

Evaluation Metrics. In addition to qualitative visual comparison, we utilize the following metrics for quantitatively evaluating the similarity between the inverted images and their original versions:

1) **Mean Square Error (MSE $\uparrow$)** [40]. It calculates the pixel-wise MSE between the inverted images and the original images. Higher MSE indicates less similarity and stronger privacy protection;

2) **Learned Perceptual Image Patch Similarity (LPIPS ↑)** [41]. LPIPS measures the perceptual image similarity between the reconstructed image features and those of ground-truth images, measured by a pre-trained VGG network. Higher LPIPS values suggest better defense;
3) **Peak Signal-to-Noise Ratio (PSNR ↓)** [42]. PSNR is the ratio of the maximum squared pixel fluctuation between two images. Lower PSNR values indicate better protection;
4) **Similarity Structural Index Measure (SSIM ↓)** [43]. SSIM evaluates the perceptual similarity between two images, considering their luminance, contrast, and structure. Lower SSIM values indicate stronger privacy protection.

B. Comparison with State-of-the-art Defenses against Diverse Attacks

In this section, we evaluate the effectiveness of our proposed defense, CENSOR, against various gradient inversion attacks and compare its performance with several leading defense baselines. The quantitative results are presented in Table I. The first column of the table lists the datasets, the second column specifies the defense methods, and the subsequent columns present the performance of these defenses against five distinct attacks. Our evaluation includes three widely-used datasets and a typical model architecture, ResNet-18 [35], in line with prior works [8], [4]. For each attack, we report four metrics, detailed in Section V-A. Here we report the initial round zero performance, as the start round is the easiest for the attacker to perform inversion and it is also the default setting in existing attack literatures [5], [9], [8], [6], [4]. These metrics provide a comprehensive assessment of inversion fidelity relative to the ground-truth images from the quantitative perspective. We also provide qualitative results in Figure 6, which visually illustrate the fidelity of the inverted images. For certain attacks that require batch normalization (BN) statistics, it is important to note that in real-world FL systems, the BN statistics derived from private data are typically not transmitted. Consequently, we do not apply the strong BN prior for these attacks, following existing works official implementations [8], [4]. Given that the randomly initialized values of inverted images significantly impact the reconstruction outcomes, following existing work [4], we perform four trials for each attack and report the attack inversion result with the best performance. In addition, for each attack, we perform the gradient inversion on ten different images for every 1000-th image of ImageNet, FFHQ and CIFAR-10 validation set. We report the numerical results for quantitative metrics are the averages of 10 inversions. We also perform an overhead evaluation in Appendix B.

Performance Against Stochastic Optimization Attacks. IG [5] and GI [9] are two typical attacks that utilize stochastic optimization without the help of GANs. It is noteworthy that our defense, CENSOR, achieves performance comparable to the state-of-the-art defense Soteria [16], and significantly surpasses other defenses. The slight superiority (ranging from 2%-18% in various metrics) of Soteria over CENSOR can be attributed to its special design of perturbed gradient. Essentially, Soteria employs stochastic optimization to invert a dummy input, utilizing the gradient computed on the dummy input to deceive the inversion process. If it can accurately approximate the attacker's inverted image and prevent it in the gradient space, it achieves success in data protection. Since Soteria is based on stochastic optimization, it is highly effective against attacks relying on stochastic optimization, such as IG and GI. However, it may fall short in GAN-based inversion scenarios, as it cannot approximate the inversion process of GANs. CENSOR, though without such a dummy input inversion, still achieves comparable quantitative results and effectively prevents attackers from inverting meaningful images, as demonstrated in the last row of Figure 6. Moreover, Soteria is less effective against more sophisticated GAN-based attacks, e.g., GGL, GIAS and GIFD. CENSOR, on the other hand, maintains its efficacy across a broader range of attacks.

Performance Against GAN-based Reconstruction Attacks. GAN-based attacks, i.e., GGL [8], GIAS [6] and GIFD [4], typically produce better reconstruction results compared to stochastic optimization. This is reasonable, as GAN-based inversion is facilitated by the pre-trained GAN to generate higher quality images. From the quantitative results in Table I, we observe that CENSOR outperforms existing defenses in almost all cases, and significantly surpasses the state-of-the-art defense Soteria (up to 114% in the metrics). We also observe that in few special cases, i.e., GGL attack on ImageNet and CIFAR-10 datasets, Clipping [14] slightly outperforms CENSOR for 14% in MSE loss and 5% in PSNR. This could potentially be reasoned as sometimes, large values in the gradient strongly reflect the data information from the original training samples, particularly in the case of GGL. In such scenarios, Clipping, which removes large values, can effectively and precisely safeguard privacy. However, such cases are rare and CENSOR is generally more effective than Clipping in all other cases. Moreover, from the visualization in the last row of Figure 6, we can observe even with the aid of GAN, it is challenging for the attacker to invert meaningful images under CENSOR.

Performance of Larger Batch Sizes. Although a batch size of 1 at each local step is the simplest setting for attackers, we evaluate the defense performance with larger (4) batch sizes. Our experiments are conducted on ImageNet using two state-of-the-art attacks, i.e., GIAS and GIFD. Notably, for these attacks, we ensure that no duplicate labels are present in each batch and they infer the labels from the received gradients [9]. We compare the defense performance of CENSOR against other baselines. The results are presented in Table II. Observe that CENSOR is robust against larger batch sizes and outperforms all other baselines across all metrics, indicating its general effectiveness. We also study different number of training rounds in Appendix C and different number of clients in Appendix D.

C. Convergence Study

We undertake a convergence study within the framework of federated learning as delineated by [1]. Our experimental evaluations are carried out using the CIFAR-10 dataset [38]

Table I: Quantitative evaluation of various defense methods against existing attacks. (An upward arrow denoting the higher the better, a downward arrow denoting the lower the better.)

DA	Defense	IG [5]				GI [9]				GGL [8]				GIAS [6]				GIFD [4]			
		MSE↑	LPIPS↑	PSNR↓	SSIM↓	MSE↑	LPIPS↑	PSNR↓	SSIM↓	MSE↑	LPIPS↑	PSNR↓	SSIM↓	MSE↑	LPIPS↑	PSNR↓	SSIM↓	MSE↑	LPIPS↑	PSNR↓	SSIM↓
ImageNet	No Defense	0.0195	0.5574	17.819	0.2309	0.0191	0.5402	17.908	0.2400	0.0453	0.5952	13.873	0.0745	0.0191	0.4795	18.452	0.3099	0.0130	0.3782	21.364	0.4528
	Noise [13]	0.0246	0.6294	16.338	0.1754	0.0269	0.6300	15.883	0.1549	0.0410	0.5697	14.252	0.0817	0.0253	0.5947	16.601	0.1854	0.0196	0.5380	18.166	0.2686
	Clipping [14]	0.0167	0.5008	18.883	0.3128	0.0383	0.7302	14.844	0.0106	**0.0477**	0.5823	**13.520**	0.0749	0.0203	0.4825	18.738	0.3186	0.0150	0.4433	19.547	0.3798
	Sparsi [15]	0.0137	0.4945	19.383	0.3419	0.0157	0.4941	18.799	0.3099	0.0456	0.6080	13.743	0.0776	0.0135	0.3981	20.483	0.4182	0.0179	0.4444	19.486	0.3686
	Soteria [16]	**0.0662**	**0.7596**	**12.220**	0.0135	**0.0682**	0.7485	**12.215**	0.0134	0.0461	0.5986	13.879	0.0708	0.0245	0.4986	17.646	0.2664	0.0139	0.3967	20.602	0.4335
	CENSOR	0.0600	0.7551	12.463	**0.0067**	0.0416	**0.8615**	14.446	**0.0021**	0.0419	**0.7912**	14.262	**0.0094**	**0.0650**	**0.7591**	**12.266**	**0.0139**	**0.0507**	**0.7610**	**13.323**	**0.0094**
FFHQ	No Defense	0.0143	0.5247	18.666	0.4209	0.0194	0.5692	17.246	0.3490	0.0421	0.5424	14.167	0.1953	0.0173	0.4228	18.738	0.4792	0.0149	0.4353	20.116	0.5102
	Noise [13]	0.0311	0.6666	15.298	0.2377	0.0388	0.7131	14.243	0.1507	0.0454	0.5784	13.866	0.1752	0.0225	0.5878	17.070	0.3449	0.0190	0.5250	17.953	0.3972
	Clipping [14]	0.0262	0.6245	15.963	0.2865	0.0593	0.8223	12.514	0.0080	0.0381	0.4881	14.571	0.2286	0.0171	0.4267	18.693	0.4808	0.0147	0.4353	20.113	0.5139
	Sparsi [15]	0.0225	0.5745	17.113	0.3411	0.0278	0.6189	16.068	0.2889	0.0421	0.5479	14.182	0.2035	0.0146	0.4057	19.490	0.5130	0.0125	0.4070	20.581	0.5423
	Soteria [16]	**0.1124**	**0.8446**	**9.662**	0.0165	**0.1105**	**0.8467**	**9.753**	0.0157	0.0438	0.5279	13.842	0.2014	0.0114	0.3560	20.551	0.5489	0.0165	0.4686	19.296	0.4789
	CENSOR	0.1009	0.8347	10.125	**0.0080**	0.0992	0.8335	10.550	**0.0061**	**0.0823**	**0.8155**	**10.947**	**0.0219**	**0.1108**	**0.7895**	**9.613**	**0.0268**	**0.1037**	**0.8097**	**9.904**	**0.0195**
CIFAR-10	No Defense	0.0023	0.0905	26.324	0.8139	0.0192	0.3909	17.170	0.4574	0.0275	0.5569	15.601	0.1099	0.0009	0.0333	30.414	0.9276	0.0201	0.5297	16.968	0.2408
	Noise [13]	0.0052	0.2229	22.810	0.7010	0.0060	0.2587	22.242	0.7241	0.0292	0.5974	15.339	0.1002	0.0022	0.2600	26.660	0.6555	0.0137	0.5166	18.648	0.3581
	Clipping [14]	0.0036	0.1050	24.396	0.8048	0.0496	0.5317	13.042	0.0482	**0.0689**	0.6125	**11.621**	0.0935	0.0014	0.0568	28.567	0.8863	0.0228	0.5492	16.426	0.2179
	Sparsi [15]	0.0093	0.3210	20.311	0.5933	0.0041	0.1631	23.841	0.7614	0.0583	0.6491	12.340	0.1407	0.0003	0.0100	35.652	0.9631	0.0297	0.4557	15.267	0.2432
	Soteria [16]	0.0558	0.5846	12.537	0.1222	**0.0891**	0.6986	**10.501**	0.0378	0.0326	0.6274	14.867	0.0731	0.0012	0.0461	29.308	0.9050	0.0239	0.5577	16.222	0.3907
	CENSOR	**0.0939**	**0.6884**	**10.272**	**0.0112**	0.0806	**0.7006**	10.937	**0.0167**	0.0260	**0.7159**	15.857	**0.0179**	**0.0789**	**0.6819**	**11.027**	**0.0789**	**0.0916**	**0.6398**	**10.381**	**0.1452**

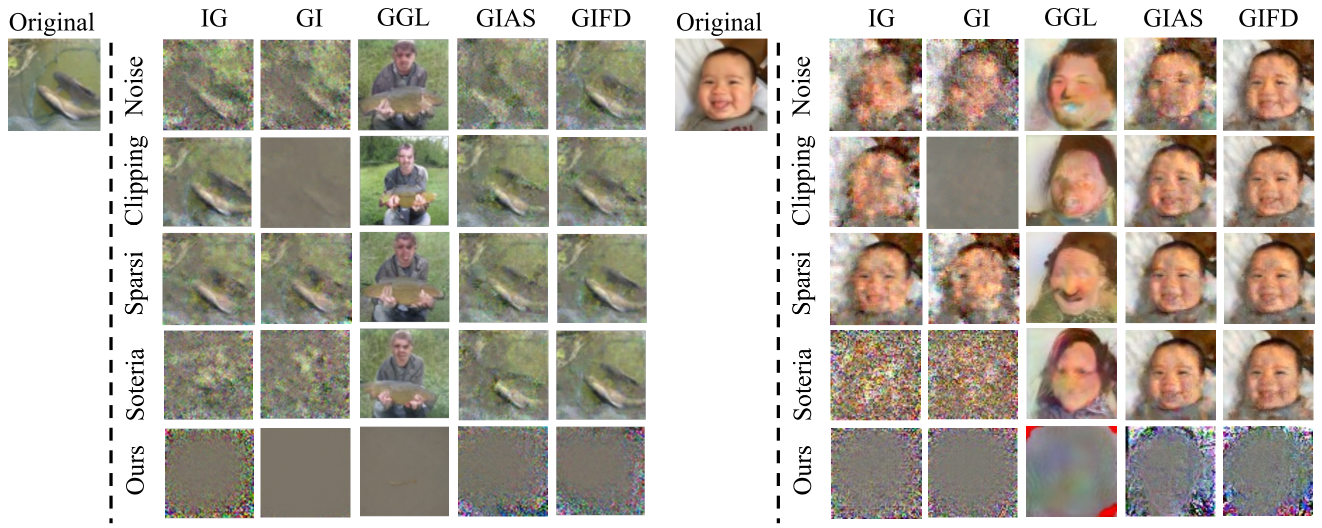

Figure 6: Qualitative evaluation of various attack inversions under existing defenses.

and employ the ResNet-18 model architecture [35]. Notably, our data are not independent and identically distributed (*non-i.i.d.*), better reflecting the complexities of real-world scenarios. This experimental design is consistent with the approach of [44], which utilizes a Dirichlet distribution [45] to simulate the *non-i.i.d.* nature of the data.

Our experimental setup involves 100 clients by default, participating in a comprehensive training process spanning 2000 rounds. In each round, we randomly selected 10 clients to participate. The training utilizes stochastic gradient descent as the optimization technique, with a local learning rate set at 0.1 and a batch size of 64. Additionally, orthogonal gradient updates are implemented on each local client, and we assess the impact of different numbers of trials on the

Table II: Evaluation for batch size of 4.

DA	Defense	GIAS				GIFD			
		MSE ↑	LPIPS ↑	PSNR ↓	SSIM ↓	MSE ↑	LPIPS ↑	PSNR ↓	SSIM ↓
ImageNet	No Defense	0.0249	0.5569	16.519	0.2096	0.0271	0.5582	16.332	0.2309
	Noise	0.0368	0.6551	14.776	0.1077	0.0350	0.6401	14.991	0.1338
	Clipping	0.0267	0.5666	16.223	0.2100	0.0294	0.5655	15.943	0.2092
	Sparsi	0.0284	0.5650	16.137	0.2128	0.0286	0.5587	16.256	0.2221
	Soteria	0.0259	0.5641	16.536	0.2037	0.0284	0.5593	16.174	0.2247
	CENSOR	**0.0776**	**0.7680**	**11.686**	**0.0099**	**0.0686**	**0.7614**	**12.245**	**0.0088**

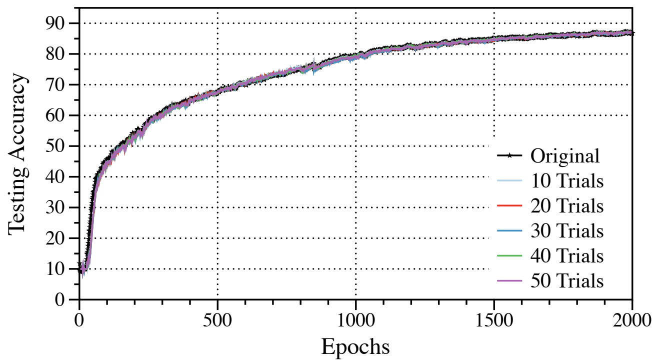

Figure 7: Testing accuracy on CIFAR-10.

orthogonal gradient selection process. Figure 7 displays the testing accuracy results. The testing accuracy across different experimental conditions reach roughly similar levels, with both standard original (vanilla) setting and adapted CENSOR's defense configurations achieving about 86% accuracy after 2000 training rounds, on the CIFAR-10 dataset under *non-i.i.d.* conditions. Notably, the discrepancies are minimal during the initial phases of training when comparing the vanilla setting with CENSOR's varying trial numbers setting. Similarly, as illustrated in Figure 8, the testing loss exhibits only slight variations between the standard original training and CENSOR defense strategy during the early stages of training. By the end of the 2000-th round, the testing loss for both configurations have settled at a relatively low level. The results indicate that CENSOR will not impact the convergence of FL model training.

D. Adaptive Attack: Expectation Over Transformation (EOT)

As attackers may devise adaptive strategies to overcome CENSOR, this section introduces a countermeasure and thoroughly evaluates CENSOR under scenarios involving adaptive attacks. Our defense method, CENSOR, is particularly designed to explore lower-dimensional manifolds within a high-dimensional neural network's parameter space, it would be very challenging for the adversary to identify and map the gradient that is applied with defense to the original gradient. To remedy this situation and construct strong adversary, we integrate state-of-the-art existing attack GIFD with the Expectation Over Transformation (EOT) [46], since EOT has been considered as

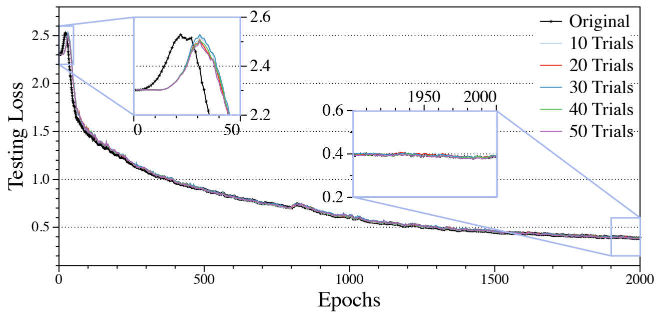

Figure 8: Testing loss on CIFAR-10.

Table III: Adaptive attack with EOT.

Dataset	EOT	MSE ↑	LPIPS ↑	PSNR ↓	SSIM ↓
ImageNet	w/o	0.0507	0.7610	13.32	0.0094
	w/.	0.0518	0.7668	13.39	0.0087
FFHQ	w/o	0.1037	0.8097	9.90	0.0195
	w/.	0.1098	0.8340	9.82	0.0195

an effective strategy to mitigate the random effect induced by the defenders [47], [48]. The idea of EOT is to perform the gradient transformation multiple times, and take the average gradient over several runs, to approximate the gradient and mitigate the randomization effect as much as possible.

Our assessment is carried out using two well-known datasets, ImageNet and FFHQ, applying EOT alongside the state-of-the-art GIFD attack [4]. The attacker follows the sampling methodology used by CENSOR's defense algorithm, generating orthogonal gradients and subsequently normalizing them in the same manner. After that, the attacker computes the average of the collected gradients over the number of trials. During this procedure, the attacker also refine the averaging process to prioritize gradients that are more informative or indicative of the original data. As demonstrated in Table III, even when subjected to the EOT-enhanced attack, CENSOR's performance remains unaffected, consistently producing significantly different inverted images compared to the original images. This indicates that our defense mechanism still prevents the attacker to invert any useful information comparing without applying EOT. This effectiveness distinctly highlights the robustness of CENSOR against adaptive attacks. The underlying reason is as discussed in the theoretical analysis in Section IV that since the orthogonal subspace of a gradient of a model with m parameters has $(m-1)$-dimensions, and the resulting orthogonal gradient is chosen from that high dimension, it would be very challenging for the adversary to identify and map the after-defense gradient to the original gradient. We also discuss other possible adaptive attacks in Appendix E.

E. Ablation Study

In this section, we conduct several ablation studies to investigate the impact of our design components and hyper-parameters. The effect of applying layer-wise operation can be found in Appendix F.

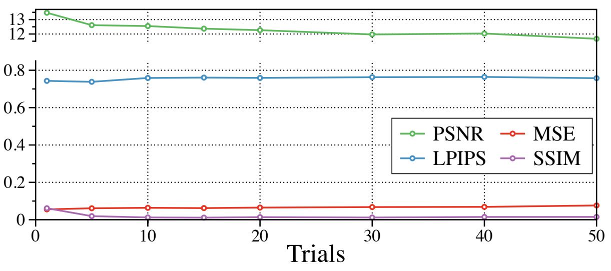

Figure 9: Different trials on ImageNet with GIAS.

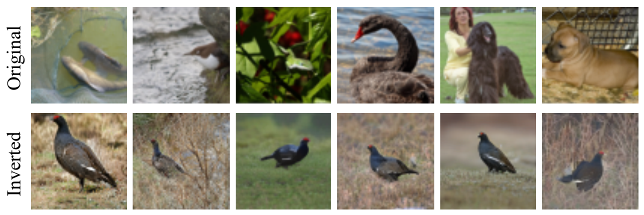

Figure 10: Gradient inversion under CENSOR, with random gradient vectors constructed from "Black grouse" images.

Effect of Different Number of Trials. In Section IV, we mention that CENSOR randomly samples $T = 20$ gradients and selects the one with the highest loss reduction. In this study, we investigate the impact of varying the number of trials. Our experiment is conducted on ImageNet using the GIAS attack under our defense. Figure 9 shows the quantitative results across four metrics. Figure 11 (in Appendix F) illustrates the inverted images for different numbers of trials. We observe that initially the metrics tend to improve with an increasing number of trials, which is expected as more trials allow CENSOR to find a more optimal gradient update. However, as the number of trials increases beyond 20, these metrics are converged, indicating that CENSOR can identify the best gradient within 20 trials and that additional trials yield plain enhancement. Therefore, we set the default number of trials to 20. Furthermore, in Figure 11, while the attacker can invert some meaningful images with 1 and 5 trials, they fail to do so with 10 and 20 trials, as evidenced by the resulting noise in the inverted images.

Effect of Sampling from a Public Dataset Other than a Normal Distribution $\mathcal{N}$. We examine the effect of sampling gradients from a publicly available dataset other than from the normal distributions $\mathcal{N}$ for CENSOR. Typically, we utilize a normal distribution for $\mathcal{N}$, which demonstrates robust performance across various datasets and attacks, as shown in Table I. In this experiment, we explore an alternative way to sample the perturbation that uses gradients derived from real images and evaluate the performance of CENSOR. The experiment is conducted on ImageNet using the GGL attack. Results, as presented in Figure 10, are based on constructing random gradient vectors from "Black grouse" images. We observe that the GGL [8] attack, which relies on GANs, is misled by our obfuscated gradients, resulting in the inversion of images that resemble black grouse, significantly different from the original training images. This demonstrates that CENSOR's flexibility of accessing orthogonal gradient in different ways, as explained in theory part of Section IV, and is able to mislead the attackers in their inversion attempts.

VI. Related Work

Subspace Learning. Several works have investigated the geometry of the loss landscape of neural networks [49], [50]. *Subspace learning* considers how neural networks can be optimized by exploring lower-dimensional subspaces within their high-dimensional parameter spaces. These subspaces are defined as the set of parameter configurations near the current parameters that still achieve similar performance levels. In particular, recent work shows there is significant freedom in the optimization paths that can achieve high accuracy models in large neural network models [17], [18], [51], [52]. Wortsman et al. [18] have notably identified and traversed large, diverse regions of the objective landscape through strategic exploration of these subspaces.

Data Privacy. In federated learning, recent studies indicate that user privacy can still be compromised in the presence of malicious attackers. Existing works on membership inference attacks [53], [54], [55], [56], [57], [58] involve a malicious entity determining whether a particular data sample was included in the training dataset. Subsequent studies have shown that the possibility of property inference attacks [59], [60]. It is furthermore explored on model inversion [61], [62], [63], [64], which elucidates the confidence scores output facilitates model inversion attacks. Researchers find that recovering a recognizable face image of a person from shallow neural networks, using only their names and the model's output confidences [61]. Further advancements in this area have demonstrated the feasibility of physical attacks [65], [66], [67].

Defenses in Data Privacy. Various defense techniques have been proposed to counter privacy attacks in federated learning. Strategies such as reducing the overfitting of the global model [54], [56], implementing differential privacy [68], [69], and masking confidence scores [54], [70] have been proven effective in mitigating the risk of membership inference. To counter reconstruction attacks, researchers [71], [72] employ multi-party computation to secure model updates. Aggregation mechanisms [73], [74], [75] are applied to defend the Byzantine attacks. Existing studies [76], [77] utilize Homomorphic Encryption to perform operations within the ciphertext space during gradient aggregation.

VII. Conclusion

We propose a novel defense technique, CENSOR, designed to mitigate gradient inversion attacks. Our method samples gradients within a subspace orthogonal to the original gradients. By leveraging cold posteriors selection, CENSOR employs a refined gradient update mechanism to enhance the data protection, while maintaining the model utility. Our experiments show that CENSOR substantially outperforms the state-of-the-art defenses, especially against advanced GAN-based attacks.

Acknowledgment

We thank the anonymous reviewers for their constructive comments. We are grateful to the Center for AI Safety for providing computational resources. This work was funded in part by the National Science Foundation (NSF) Awards IIS-2229876, CNS-2247794, CAREER IIS-1943364, CNS-2212160, SHF-1901242, SHF-1910300, IIS-2416835, IARPA TrojAI W911NF-19-S0012, ONR N000141712045, N000141410468 and N000141712947. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang, "Secureboost: A lossless federated learning framework," *IEEE Intelligent Systems*, vol. 36, no. 6, pp. 87–98, 2021.

[3] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Federated learning for ultra-reliable low-latency v2v communications," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[4] H. Fang, B. Chen, X. Wang, Z. Wang, and S.-T. Xia, "Gifd: A generative gradient inversion method with feature domain optimization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4967–4976.

[5] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 937–16 947, 2020.

[6] J. Jeon, K. Lee, S. Oh, J. Ok *et al.*, "Gradient inversion with generative image prior," *Advances in neural information processing systems*, vol. 34, pp. 29 898–29 908, 2021.

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

[8] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 132–10 142.

[9] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 337–16 346.

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

[11] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[12] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning: Revisited and enhanced," in *Applications and Techniques in Information Security: 8th International Conference, ATIS 2017, Auckland, New Zealand, July 6–7, 2017, Proceedings*. Springer, 2017, pp. 100–110.

[13] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[14] W. Wei, L. Liu, Y. Wu, G. Su, and A. Iyengar, "Gradient-leakage resilient federated learning," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2021, pp. 797–807.

[15] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 440–445.

[16] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. 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*, 2021, pp. 9311–9319.

[17] A. Chaudhry, N. Khan, P. Dokania, and P. Torr, "Continual learning in low-rank orthogonal subspaces," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9900–9911, 2020.

[18] M. Wortsman, M. C. Horton, C. Guestrin, A. Farhadi, and M. Rastegari, "Learning neural network subspaces," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 217–11 227.

[19] P. Izmailov, S. Vikram, M. D. Hoffman, and A. G. G. Wilson, "What are bayesian neural network posteriors really like?" in *International conference on machine learning*. PMLR, 2021, pp. 4629–4640.

[20] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[21] L. Fowl, J. Geiping, W. Czaja, M. Goldblum, and T. Goldstein, "Robbing the fed: Directly obtaining private data in federated learning with modified models," *arXiv preprint arXiv:2110.13057*, 2021.

[22] Y. Wen, J. Geiping, L. Fowl, M. Goldblum, and T. Goldstein, "Fishing for user data in large-batch federated learning via gradient magnification," *arXiv preprint arXiv:2202.00580*, 2022.

[23] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "Gan inversion: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 3121–3138, 2022.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[25] A. Hatamizadeh, H. Yin, H. R. Roth, W. Li, J. Kautz, D. Xu, and P. Molchanov, "Gradvit: Gradient inversion of vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 021–10 030.

[26] G. Daras, J. Dean, A. Jalal, and A. G. Dimakis, "Intermediate layer optimization for inverse problems using deep generative models," *arXiv preprint arXiv:2102.07364*, 2021.

[27] K. Wei, J. Li, M. Ding, C. Ma, H. Su, B. Zhang, and H. V. Poor, "User-level privacy-preserving federated learning: Analysis and performance optimization," *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3388–3401, 2021.

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

[29] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora, "Evaluating gradient inversion attacks and defenses in federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7232–7241, 2021.

[30] C. M. Bishop, "Pattern recognition and machine learning," *Springer google schola*, vol. 2, pp. 645–678, 2006.

[31] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.

[32] R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson, "Cyclical stochastic gradient mcmc for bayesian deep learning," *arXiv preprint arXiv:1902.03932*, 2019.

[33] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal gradient descent for continual learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 3762–3773.

[34] T. Doan, M. A. Bennani, B. Mazoure, G. Rabusseau, and P. Alquier, "A theoretical analysis of catastrophic forgetting through the ntk overlap matrix," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 1072–1080.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[37] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

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

[39] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[40] Y. Dodge, *The concise encyclopedia of statistics*. Springer Science & Business Media, 2008.

[41] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in

Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 586–595.

[42] W. Lin, L. Dong, and P. Xue, "Visual distortion gauge based on discrimination of noticeable contrast changes," *IEEE transactions on circuits and systems for video technology*, vol. 15, no. 7, pp. 900–909, 2005.

[43] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

[44] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 2938–2948.

[45] T. Minka, "Estimating a dirichlet distribution," 2000.

[46] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 284–293.

[47] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 274–283.

[48] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," *Advances in neural information processing systems*, vol. 33, pp. 1633–1645, 2020.

[49] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," *Advances in neural information processing systems*, vol. 31, 2018.

[50] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of dnns," *Advances in neural information processing systems*, vol. 31, 2018.

[51] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, "Qualitatively characterizing neural network optimization problems," *arXiv preprint arXiv:1412.6544*, 2014.

[52] P. Izmailov, W. J. Maddox, P. Kirichenko, T. Garipov, D. Vetrov, and A. G. Wilson, "Subspace inference for bayesian deep learning," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 1169–1179.

[53] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[54] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[55] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1897–1914.

[56] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv:1806.01246*, 2018.

[57] Z. Li and Y. Zhang, "Membership leakage in label-only exposures," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 880–895.

[58] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, "Membership privacy: a unifying framework for privacy definitions," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 889–900. [Online]. Available: https://doi.org/10.1145/2508859.2516686

[59] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 619–633. [Online]. Available: https://doi.org/10.1145/3243734.3243834

[60] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 691–706.

[61] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[62] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 253–261.

[63] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, "Neural network inversion in adversarial setting via background knowledge alignment," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 225–240.

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

[65] S. An, Y. Yao, Q. Xu, S. Ma, G. Tao, S. Cheng, K. Zhang, Y. Liu, G. Shen, I. Kelk *et al.*, "Imu: Physical impersonating attack for face recognition system with natural style changes," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 899–916.

[66] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.

[67] S. An, L. Yan, S. Cheng, G. Shen, K. Zhang, Q. Xu, G. Tao, and X. Zhang, "Rethinking the invisible protection against unauthorized image usage in stable diffusion," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 3621–3638.

[68] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 2018, pp. 268–282.

[69] B. Hui, Y. Yang, H. Yuan, P. Burlina, N. Z. Gong, and Y. Cao, "Practical blind membership inference attack via differential comparisons," *arXiv preprint arXiv:2101.01341*, 2021.

[70] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 259–274.

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

[72] D. Lia and M. Togan, "Privacy-preserving machine learning using federated learning and secure aggregation," in *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, 2020, pp. 1–6.

[73] H. Guo, H. Wang, T. Song, Y. Hua, Z. Lv, X. Jin, Z. Xue, R. Ma, and H. Guan, "Siren: Byzantine-robust federated learning via proactive alarming," in *Proceedings of the ACM Symposium on Cloud Computing*, 2021, pp. 47–60.

[74] K. Zhang, G. Tao, Q. Xu, S. Cheng, S. An, Y. Liu, S. Feng, G. Shen, P.-Y. Chen, S. Ma, and X. Zhang, "FLIP: A provable defense framework for backdoor mitigation in federated learning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=Xo2E217_M4n

[75] H. Guo, H. Wang, T. Song, Y. H. R. Ma, X. Jin, Z. Xue, and H. Guan, "Siren+: Robust federated learning with proactive alarming and differential privacy," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[76] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE transactions on information forensics and security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[77] J. Kim, D. Koo, Y. Kim, H. Yoon, J. Shin, and S. Kim, "Efficient privacy-preserving matrix factorization for recommendation via fully homomorphic encryption," *ACM Transactions on Privacy and Security (TOPS)*, vol. 21, no. 4, pp. 1–30, 2018.

Appendix

A. *Summary of Symbols*

We summarize a comprehensive list of all notations in Table IV for easy reference.

Table IV: Glossary of Notations.

Notation	**Description**
f	Model
θ	Model parameter
$\hat{\theta}$	optimal model parameter
ℓ	Loss function
$\mathbb{R}^d$	Input space
$\mathbb{R}^k$	Latent space of the generative model
N	Number of local clients
k	k-th local client
n	Batch size on the client
τ	Number of federated learning iteration
X', Y'	Randomly initialized inputs
$\hat{X}', \hat{Y}'$	Reconstructed inputs
$\mathcal{D}(\cdot)$	Distance metric
$G_p(\cdot)$	Pre-trained generative model
$\mathcal{T}(\cdot)$	Gradient transformation function
$\phi(\cdot)$	Regularization term
D	Dataset
$P(\cdot\|\cdot)$	Normalized probabilities
M	Temperature paramete
T	Sampling trials
$\mathcal{N}(\mathbf{0}, \cdot)$	Normal distribution
$\perp$	Orthogonal
$\langle \cdot, \cdot \rangle$	Inner product
G^0	Original gradient
$G^\star$	Best gradient
z	Latent space of the generative model
C	Number of classes
W_{FC}	Weights of final fully-connected (FC) classification layer
g^o	Orthogonal gradients
g_r	Random gradients
l	Layer

B. *Overhead Evaluation*

In this section, we conduct an evaluation to quantify the computational overhead and additional processing latency of CENSOR and Soteria [16]. We define the overhead percentage as: Overhead Percentage = ((Time with Defense - Time without Defense) / Time without Defense)$\times$100%. Applying this formula to CENSOR, the overhead calculation yields an exceptionally low impact of approximately 0.00236%. Comparatively, we also assessed Soteria, which introduces an overhead of approximately 0.3092%, significantly higher than CENSOR.

C. *Performance of* CENSOR *Applied for Different Number of Training Rounds.*

We investigate the impact of applying CENSOR across various numbers of training rounds to assess its performance under different conditions. This experiment is conducted using the ImageNet dataset and the GIFD attack. We systematically apply CENSOR from 0 to 9 training rounds and measure the performance of the attack at each interval. The results of this study are detailed in Table V, where we present four quantitative metrics for each training round to provide a comprehensive evaluation. According to the findings presented in Table V, CENSOR consistently demonstrates effectiveness in countering the GIFD attack through various stages of training. Notably, as the number of training rounds increases from 0 to 9, we observe a noticeable increase in the MSE loss from 0.0507 to 0.1034, while the PSNR decreases from 13.32 to 10.65, it still outperforms existing defenses. This improvement in defensive capability against the GIFD attack not only highlights the robustness of CENSOR but also corroborates our earlier observation documented in Section III. Specifically, as the training progresses and the model approaches convergence, it becomes progressively more difficult for an attacker to successfully reconstruct raw data from the gradients.

Table V: Ablation study for different number of rounds.

Round ID	**GIFD**			
	MSE $\uparrow$	**LPIPS $\uparrow$**	**PSNR $\downarrow$**	**SSIM $\downarrow$**
0	0.0507	0.7610	13.32	0.009
1	0.0989	0.7368	10.50	0.036
2	0.1224	0.7394	9.94	0.036
3	0.1040	0.7370	10.37	0.042
4	0.1717	0.7528	9.07	0.041
5	0.1233	0.7296	10.16	0.035
6	0.1049	0.7272	10.28	0.034
7	0.1232	0.7425	9.95	0.033
8	0.1067	0.7173	10.69	0.041
9	0.1034	0.7247	10.65	0.038

D. *Performance of* CENSOR *Across Different Number of Clients.*

We evaluate the performance of CENSOR in different numbers of participating clients. Specifically, our experimental setup involves 100 clients by default. and we test our defense with 10, 30, and 50 selected clients out of 100 on the CIFAR-10 dataset under a *non-i.i.d.* setting. Results in Table VI show that our technique consistently converges and matches the performance of the vanilla aggregated model without defense, regardless of the number of clients. This suggests that our defense scales effectively with the number of clients and does not necessarily lead to a substantial decrease in performance.

Table VI: Performance of CENSOR across different number of clients.

Number of Clients	Accuracy
10	85.27
30	85.33
50	85.30

E. *Discussion of Adaptive Attacks*

In this section, we discuss additional possible adaptive attacks. First, leveraging a similar mechanism as CENSOR to train a Generative Adversarial Networks (GAN) and recover the original gradients. Existing GAN models [37], [39] utilize considerably smaller latent spaces, although it is non-trivial to employ the gradient as a latent space to train a GAN to reconstruct the original gradients effectively in the context of

CENSOR, it can be explored in the future. Second, given an overparameterized model, there exists many different directions that can improve the loss. If the attacker has knowledge about the cold posterior sampling mechanism, they might attempt to manipulate the temperature parameter. By injecting specially crafted updates that influence the posterior distribution, attackers could bias the sampling process, making it easier to infer the original gradients.

F. Ablation Study

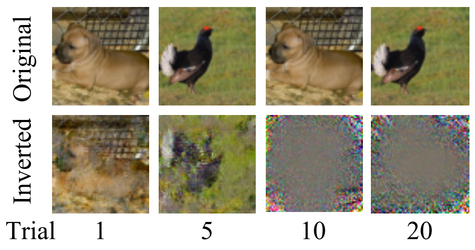

Figure 11: Trials examples on GIAS.

Effect of Applying Layer-wise Operation. We explore the impact of applying the layer-wise operation of CENSOR. The experiment, conducted on ImageNet using the GIFD attack, is detailed in Table VII. We assess the inversion performance of the attack using four metrics. Results indicate that the layer-wise application of CENSOR slightly outperforms the technique when applied to the entire gradient vector of the whole model. This improvement can be attributed to the fine-grained orthogonal projection and normalization processes detailed in Section IV, which enhance the specificity and efficacy of the defense by adapting it more precisely to the unique characteristics of each layer.

Table VII: Ablation study on layer-wise operation.

Config	MSE ↑	LPIPS ↑	PSNR ↓	SSIM ↓
Layer-wise gradient	**0.0507**	**0.7610**	**13.32**	**0.009**
Entire gradient	0.0452	0.7532	13.81	0.012