Closing the Loophole: Rethinking Reconstruction Attacks in Federated Learning from a Privacy Standpoint

Seung Ho Na, Hyeong Gwon Hong, Junmo Kim, and Seungwon Shin
{harry.na, honggudrnjs, junmo.kim, claude}@kaist.ac.kr
KAIST, Republic of Korea

ABSTRACT

Federated Learning was deemed as a private distributed learning framework due to the separation of data from the central server. However, recent works have shown that privacy attacks can extract various forms of private information from legacy federated learning. Previous literature describe differential privacy to be effective against membership inference attacks and attribute inference attacks, but our experiments show them to be vulnerable against reconstruction attacks. To understand this outcome, we execute a systematic study of privacy attacks from the standpoint of privacy. The privacy characteristics that reconstruction attacks infringe are different from other privacy attacks, and we suggest that privacy breach occurred at *different levels*. From our study, reconstruction attack defense methods entail heavy computation or communication costs. To this end, we propose Fragmented Federated Learning (FFL), a lightweight solution against reconstruction attacks. This framework utilizes a simple yet novel gradient obscuring algorithm based on a newly proposed concept called the global gradient and determines which layers are safe for submission to the server. We show empirically in diverse settings that our framework improves practical data privacy of clients in federated learning with an acceptable performance trade-off without increasing communication cost. We aim to provide a new perspective to privacy in federated learning and hope this privacy differentiation can improve future privacy-preserving methods.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Security and privacy**;

KEYWORDS

Reconstruction Attack, Data Privacy, Federated Learning

ACM Reference Format:
Seung Ho Na, Hyeong Gwon Hong, Junmo Kim, and Seungwon Shin. 2022. Closing the Loophole: Rethinking Reconstruction Attacks in Federated Learning from a Privacy Standpoint. In *Annual Computer Security Applications Conference (ACSAC '22), December 5–9, 2022, Austin, TX, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3564625.3564657

1 INTRODUCTION

Various incidents of data compromise made data privacy an important issue in deep learning. Traditional machine learning models have training and testing carried out in the same machine, and the model will use data for learning on the same machine. To incorporate data privacy to deep learning, federated learning (FL) emerged as a private data learning framework by separating data storage and actual model learning [33].

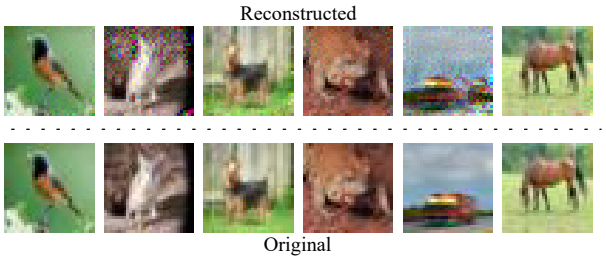

Figure 1: Reconstruction attacks on differentially private models [23] of CIFAR-10. Models were trained under the privacy budget of $\epsilon = 8$.

This seemingly secure framework, however, holds vulnerabilities to *privacy attacks*: attacks aiming at leaking private information [25, 53, 72]. Although the server does not have direct access to the clients' data, the model may leak information regarding membership or unseen attribute information [21, 44, 49, 56, 58]. The most critical form of attack utilizing these vulnerabilities are *reconstruction attacks*, in which previous studies have shown that from the gradient information, train data of the model can be reconstructed [22, 64, 72]. According to these studies, if the central server were to be malicious with the intent on compromising the privacy of its clients, it could do so without direct access to the data and rely on optimization techniques to backtrack and reconstruct the private data.

Due to these threats extracting private information, many privacy-preservation techniques were developed. A prominent privacy-preserving method is differential privacy. Differential privacy [16] is a theoretical approach to quantifying information leakage and offers privacy guarantees. In differential privacy, each client's privacy is preserved by adding noise to sensitive attributes. Differential privacy is known to be an effective form of defense against membership inference attacks [44, 70]. On the other hand, reconstruction attacks on differentially private models are shown to be successful (Figure 1), explained in Section 5.

Owing to this contrasting result, we execute a systematic study on privacy attacks focusing on the privacy aspect: *are different forms of privacy attacked?* In our study, we dissect privacy attacks and identify the different levels of privacy into two classes: *disclosure privacy* and *distinctive privacy*. Disclosure privacy describes the right that unintended information must not be leaked from the collaborative result, and distinctive privacy describes the right to keep raw data undisclosed (presented in Section 2.3).

By investigating this differentiation of privacy, we find that although existing privacy-preserving methods can preserve disclosure privacy, existing techniques enforcing distinctive privacy incur heavy computation or communication overhead. Examples are secure multi-party computation protocols that use key encryption

schemes to provide privacy [27, 61, 66]. Furthermore, reduction of communication cost is an important factor in FL. Literature on communication cost explore faster training (convergence in fewer number of iterations) [41, 43, 63, 69] and compression of transmitted data [6, 34, 38, 51, 54].

Establishing this as motivation, we propose the problem of *obscuring client gradients*, specifically designed to preserve distinctive privacy while maintaining low communication cost for practical FL. As a solution to this problem, we propose Fragmented Federated Learning (FFL), a light FL framework designed to preserve distinctive privacy by employing a gradient obscuring algorithm on the calculated gradient of each client.

FFL is adjustable, by allowing a hyperparameter r to manage between the defense capability of the obscured gradient and the resulting model performance. In the most conservative settings of r, FFL shows to be the most effective in preventing reconstruction attacks when compared to differential privacy [23] and gradient compression [38]. FFL also bears no additional communication cost compared to the general concept of FL. The clients only undergo a similarity calculation cost of $O(n)$, where n is the size of the model. This overhead, compared to the inevitable training calculations of the model, can be deemed trivial.

Our contributions are as follows:

- We conduct a holistic study by dissecting privacy attacks in FL. From this study, we deduce and introduce two divisions of privacy in FL: disclosure privacy and distinctive privacy.
- We classify existing privacy-preserving techniques by the form of privacy they preserve, and reveal the shared problems of distinctive-privacy-preserving techniques and adopt this as our problem-of-interest.
- We propose a new problem of *obscuring client gradients* addressing reconstruction attacks by gradient and propose FFL, a practical FL framework that utilizes fragmentation operations to preserve distinctive privacy, as a proper solution to the problem.

2 BREAKDOWN OF PRIVACY IN FEDERATED LEARNING

2.1 Federated Learning (FL)

Federated learning is a machine learning technique that trains a model across many clients with their respective data to guarantee data privacy [33, 34]. The central server provides its clients with the training model, and clients calculate the gradient in their own local environment using their private data; each client's data are not shared with the central server. The clients upload the update information to the central server, where the server aggregates this information and updates the model. This process is repeated.

In the distributed setting of FL, the update information is the calculated gradient information of each client. The aggregation of the N client gradients and the model (θ) update is indicated in Equation (1), a procedure called federated stochastic gradient descent. $\nabla_\theta L_{\theta^k}(x_i, y_i)$ refers to the gradient calculated on the model θ using the training data (x_i, y_i) of client i at step k.

$$\theta^{k+1} = \theta^k - \gamma \sum_{i=1}^{N} \nabla_\theta L_{\theta^k}(x_i, y_i) \tag{1}$$

FL is known for this separation of training and model update; the location where the model is trained is the client's local environment, whereas the location of the model update is the central server. This separation of data in FL maintains the client's private data to be remote and unreachable by the central server. Accordingly, FL has been implemented in fields handling sensitive data (e.g., loan risk, medical imaging, financial payments [28, 30, 68]).

2.2 Dissection of Privacy Attacks

In this section, we present a breakdown of privacy attacks in FL and reconsider reconstruction attacks from the standpoint of privacy. Privacy attacks are designed on extracting undisclosed information of the data that a target machine learning model was trained on. Privacy attacks in FL are applicable by both client and server of FL. The three forms of privacy attacks considered are membership inference attacks, attribute inference attacks, and reconstruction attacks and they will be examined under the three attributes of *extraction extent*, *transferability*, and *source*. Note that privacy attacks in FL are a subset of inference attacks [40], where model stealing is excluded due to its irrelevance to client privacy.

Extraction Extent: Depending on the privacy breach, the content of the leaked information may differ. In this context, we determine the actual data used in training (ultimately the most private information of an individual) as the *raw data*, and any other form of private information to be private *meta-information*.

Transferability: A privacy attack is *transferable* if the attack can be applied to other data units without change (a single model used for multiple attacks); it is *intransferable* if the attack must be conducted (retrained) for each instance.

Source: To extract information, privacy attacks must take advantage of an information source. This refers to the object or source of private data leakage: the object to secure for attack prevention. The location of this source is the entry point of the attacker.

2.2.1 Membership Inference Attacks (MIA). A form of privacy attack that aims to determine the participation of a data sample in training is referred to as a membership inference attack (MIA) [56]. The information of a data point's participation in training can lead to derivations of an individual's private information. For example, by attacking machine learning models trained on medical data such as drug dose prediction, an attacker could deduce health information of a participant. Although an authentic infringement of privacy, the extraction extent of MIA do not reach the level of raw data. Membership information is a form of private meta-information. Note that this leak of private meta-information through MIA are viewed as gateways to further attacks [13], and that we are not depreciating the breach in privacy by membership inference.

As an extensively studied field of attack, the methods take advantage of the classifier's generalization gap on the train and test data [53, 56, 70]. Attackers determine this gap by gathering auxiliary datasets (called the shadow datasets) that best mimic the distribution of the train data and test data. Once trained on the shadow datasets, attacks can be carried out on data pieces to predict

Table 1: FL privacy attacks and their attributes.

Privacy Attack	Description	Extraction Extent	Transferability	Source
MIA	from model parameters θ, leak membership info. of data point x	Meta-info.	Transferable	θ
AIA	from model parameters θ, leak attribute info. of data point x	Meta-info.	Transferable	θ
Recon. Attack	from gradient ∇ and model parameters θ, reconstruct data point x	Raw data	Intransferable	∇

membership. Because a membership inference model can be reused without change for different data pieces, they are transferable.

MIA utilize the different behavior of the model when input train and test data. Most methods use the distribution differences of the logit information of these respective data [56, 59, 70], with some approaches using additional information such as the mid-network latent information for richer context [36, 49]. Nevertheless, the source in MIA are model weights. Any node with possession of the model weights are capable of these attacks; membership information can be extracted by all clients and even the server.

2.2.2 Attribute Inference Attacks (AIA). The goal of attribute inference attacks are to exploit unseen attributes of the data that is unrelated to the original task, often called the re-purposing of a model. Previous studies were able to infer the race information of human photos from a model trained with the task of predicting age [44, 58]. Similar to MIA, attribute inference attacks (AIA) aim to leak not the specific train data but a different form of private information, so the extraction extent is at meta-information level.

AIA are also transferable attacks. The training of an attribute inference model requires a black box oracle to build an auxiliary dataset that maps the the embedding through the target model with the attribute-in-interest [58]. Once the attack model is trained for re-purposing, this model can be used to discover the wanted attribute information from target data points. With respect to the same attribute information AIA are transferable; extracting information of a different attribute requires training of a new AIA.

Two main components that make AIA plausible are the existence of an oracle for auxiliary dataset generation and the model parameters. Despite the fact that these two components are both crucial, the oracle is an item that is irrelevant with the FL procedure. As the possession/prevention of an oracle is out-of-scope, the target model is the source of an AIA. As a matter of fact, defense methods for attribute inference methods involve making the model more robust–censoring representations by employing mini-max games or information-theoretical optimizations [11, 17, 47]. These algorithms encode data points into embeddings that do not reveal unwanted attributes, yet sufficiently representative for proper functionality. Similar to membership inference attacks, AIA can be conducted in both server and client.

2.2.3 Reconstruction Attacks. Despite the private impression of FL, recent studies have exposed its vulnerability to reconstruction attacks [22, 64, 72]. The objective of reconstruction attacks is to reconstruct the original data from the description and auxiliary information of the model [20] and therefore the extraction extent is raw data. The problem of data reconstruction is a more difficult and threatening attack compared to the retrieval of membership or attribute information.

Reconstruction attacks specific to FL recover the input image by minimizing the distance between the input gradient and the sample image gradient. The works differ in the selected metric of distance: Euclidean matching [64, 72] or cosine similarity [22], with cosine similarity being more successful. Equation (2) is the objective function used for reconstruction by cosine distance, where x^* refers to the original image. The regularization term TV stands for total variation [42] and encourages spatial smoothness to the image. The optimization problems can be solved by an L-BFGS solver [39], but Geiping et al. [22] improve reconstruction quality using ADAM [32] and qualitatively demonstrated the successful reconstruction of training data. Equation 2 needs to be re-optimized for a different data piece x (different attack instance), so reconstruction attacks are intransferable.

$$\arg\min_x 1 - \frac{\langle \nabla_\theta L_{\theta^k}(x,y), \nabla_\theta L_{\theta^k}(x^*,y)\rangle}{\|\nabla_\theta L_{\theta^k}(x,y)\|\|\nabla_\theta L_{\theta^k}(x^*,y)\|} + \alpha TV(x) \qquad (2)$$

The two materials for reconstruction attacks are the model parameters and the target gradient. According to [22, 72] however, reconstruction attacks were successful even when using *randomly-initialized* model weights and the target gradient. This signified that although using trained models did show better reconstruction performance, it may not be required in reconstruction optimization. This demonstrates that reconstruction attacks are target-gradient-centric and that the target gradient is the source of information leakage. The location of reconstruction attacks are therefore unique to the server (the owner of the gradient is a victim and cannot be the attacker).

Table 1 summarizes the privacy attacks by their attributes. From this dissection of privacy attacks in FL by their attributes, we can conclude that *reconstruction attacks need to be considered differently.*

2.3 Differentiation of Privacy

This isolation of reconstruction attacks from MIA and AIA signifies contrasting forms of privacy infringement. By backtracking the attributes of privacy attacks, the agenda of privacy attacks can be represented into two planes of privacy shown in Figure 2.

The upper layer of attacks leak private meta-information sourced from the resulting model of the distributed system. In a central distributed system (e.g. FL), the service provider holds the responsibility of ensuring privacy of the clients. Specifically, the clients' private information should not be able to be deduced from any resulting outcome of the system (e.g. trained model). We label this form of privacy that the service provider bestows on the collaborative result as **disclosure privacy**. In FL, providing disclosure privacy would strengthen the robustness of the resulting *model weights* from privacy attacks.

Reconstruction attacks take advantage of already leaked private meta-information to access the private raw data. Let us take a conceptual analogy of this lower level privacy attack. Assume that password information of clients (private meta-information) of an electronic mail service were leaked. An attacker could use

this password information to access the mail contents (private raw data). Unlike common users who would be vulnerable to this reconstruction attack, a client concerned with their privacy could have been periodically updating their password for enhanced security and prevent this attack by making the leaked meta-information obsolete. Like this particular user's proactive practice, private meta-information should not be distinctive enough to characterize private raw data. Preserving **distinctive privacy** is the act of guarding private meta-information to prevent further raw data leaks. Specifically in FL, providing distinctive privacy would strengthen the robustness of the *gradient* so that reconstruction attacks fail.

The main contrast between disclosure privacy and distinctive privacy arise from knowledge of the victim. Disclosure privacy and their attacks are not associated with ownership of the data. Private meta-information from these attacks, in the case it is leaked, cannot be identified to be a certain clients' information, and neither do they intend on identifying *who's* privacy was breached. On the other hand, distinctive privacy and their attacks are targetted at the owner of the gradient information. Distinctive privacy acts to prohibit this form of targeted privacy leak of the specific individual.

But isn't disclosure privacy enough? Following the privacy attack agenda of Figure 2, if disclosure privacy was preserved, there would be no leak of private meta-information nullifying reconstruction attacks. In other words, if disclosure privacy is preserved, distinctive privacy is naturally preserved as well. But in the exclusive case of FL, disclosure privacy and distinctive privacy observe to be orthogonal due to an attacker that can evade disclosure privacy protection and manage to access private meta-information (gradient information). This is the case of an *honest-but-curious server*.

Honest-But-Curious Server: Similar to the characterization of honest-but-curious servers in various works [8, 9, 50], this setting describes a server that takes advantage of the client's private data (gradients) and exploits it without disrupting the overall protocol (collaborative learning algorithm). This means that the server may not modify or swap the model parameters to malicious parameters better suited for their attack, and the server should faithfully perform its duty as the central server and properly update and distribute the model weights after aggregating the client gradients. For the rest of the paper, the honest-but-curious server will be referred to as the *attacker*; any instance of the word *server* will refer to a general honest server.

By participating in FL, the attacker will fulfill its duty as the server and partake in the proper training of the model. In this process, it will collect gradient information directly from the clients, and therefore private meta-information will be exposed to the attacker despite any disclosure privacy preservation attempts–the attacker is not manipulating the model weights in any sort. This is a systemic loophole created by the position of an honest-but-curious server; the server is bound to collect the gradient information of clients. Distinctive privacy attacks are internal attacks being launched from inside the boundaries of disclosure privacy (Figure 2 shows that the reconstruction attacker is placed within the bounds of disclosure privacy). Note that the attacker will not only receive gradient information, it will know the ownership of this private meta-information. Therefore, the attacker holds potential of targeting and threatening each and every client of their distinctive privacy.

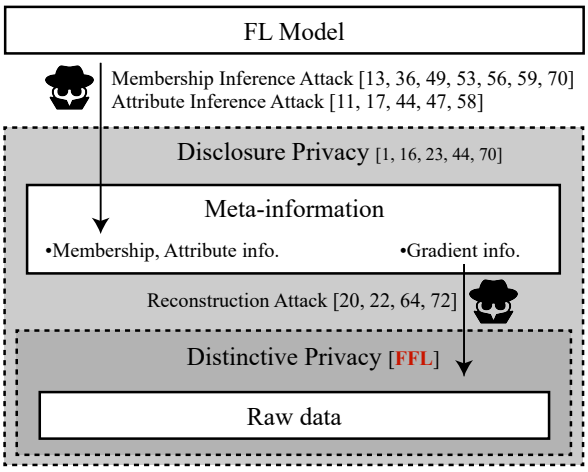

Figure 2: Breakdown of privacy attacks. To the right are citations that refer to the respective concept. We propose FFL for preserving distinctive privacy.

2.4 Privacy Preserving Methods

The privacy concerns of FL has caused the development of many techniques to improve privacy of the trained models. The most representative types of privacy preserving methods are differential privacy and secure multiparty computations.

2.4.1 Differential Privacy (DP). Differential privacy [16] is a theoretical approach to quantifying information leakage and offers privacy guarantees. It employs a randomized mechanism into the learning process and is defined as such:

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

$$Pr[\mathcal{M}(d) \in S] \leq e^{\epsilon} Pr[\mathcal{M}(d') \in S] + \delta \tag{3}$$

Differential privacy guarantees that the probability of harm will not be significantly increased by one's choice to participate in the random mechanism. Hence, given the setting of (ϵ, δ), differential privacy guarantees the result (trained model) of the random mechanism will be the same whether a data piece is included or not. Conceptually, differential privacy is a method that secures the model from leaking information on the respective data piece's contribution (private meta-information). Coincidentally, membership inference attack works discuss differential privacy as a possible defense mechanism [44, 70].

The implementation of differential privacy does include modification to the gradient information. This distortion consists of clipping to satisfy the sensitivity of Equation 3 and adding random noise [1, 23]. The purpose of this step, however, is not specifically to discourage the optimization of reconstruction attacks and is insufficient in doing so (Figure 1), evaluated later in Section 6. Differential privacy cannot be a form of preserving distinctive privacy.

2.4.2 SMC Protocols. Works incorporating secure multi-party computation (SMC) protocols use key encryption schemes to prevent

leaking of additional knowledge and provide strong privacy guarantees, but incur heavy computation and communication costs due to the overhead caused by the added complication by the protocol [27, 61, 66]. These schemes pose an overhead that limits the scalability of SMC frameworks for FL, and also are unsuitable for dealing with the flexible nature of real world application clients where the number of clients fluctuate depending on uncontrollable factors such as poor connectivity or low power [57].

Regarding this communication burden, many studies investigated methods of reducing communication cost. Largely explored methods elicit higher quality update information and train the model in a fewer number of iterations [41, 43, 63, 69] or compress the data to be transmitted in communication and scale down the total transfer size in FL [6, 34, 38, 51, 54]. Specifically for SMC, SAFELearn [19] provides efficient secure aggregation, requiring 2 rounds of communication for each training iteration (minimal considering previous SMC works).

By encrypting transmitted information (e.g. gradient information), they preserve distinctive privacy by preventing access to gradient information. Despite the many efforts of minimizing additional computation and communication overhead, due to the nature of encryption, one training round inevitably requires multiple round of communication. Conceptually similar to SMC protocols in terms of isolation of private data, works using trusted execution environments (TEE) for secure FL also isolate the attacker from gaining access to gradient information [46] without additional communication rounds. Unfortunately, TEE suffers from major latency issues caused by its cryptographic authentication protocol [52].

By differentiating the privacy of FL, we cross-check privacy-preserving methods and find that we *require a light means of preserving distinctive privacy.*

3 PROBLEM STATEMENT

3.1 Problem Formulation

Acknowledging this dilemma of distinctive privacy being accompanied with excessive communication and computation costs, we wish to explore a concise means to preserve distinctive privacy. Viewing reconstruction by gradient as the attack-of-interest, our work investigates the threat model of an *honest-but-curious* server as the adversary/attacker described in Section 2.3.

Given the threat model for reconstruction attacks, to preserve distinctive privacy, clients in FL must protect their gradient information from the attacker using its provided *allocations*. It is crucial for any defense mechanism to work under the core FL scheme when considering the communication cost and applicability of the solution. Therefore, the defense should be devised by the already provided allocations to the client. A client's allocations can be defined as below.

Definition 2 (Allocations of FL client)**.** A client will receive and therefore have access to the FL model parameters θ^k at training round k. The set of these parameters $\mathcal{A} = \{\theta^k : k \geq 1\}$ are the *allocations* of an FL client.

Using these conventional allocations that are the byproduct of FL training rounds, a client must devise a function to protect its gradients. In addition to reconstruction attack defense capability, the mechanism of defense should refrain from increasing communication costs. Like any security application, there exists an inevitable trade-off between performance and privacy protection levels [65]. If tradeoff is unavoidable, there should at least be a means to control this; privacy protection with respect to model performance should be able to be strengthened or relaxed. These requirements lead to our problem definition below:

Problem 1 (Obscuring Client Gradients)**.** Let $\nabla_\theta L_{\theta^k}(x, y)$ be the gradient calculated from the given neural network parameters θ^k and private labeled data (x, y). Obscuring client gradients is to find a function f that when input the gradient $\nabla_\theta L_{\theta^k}(x, y)$ and the allocations $\mathcal{A}$, returns an *obscured* gradient that satisfies the following conditions:

- **(c1)** $X(Recon(f(\nabla_\theta L_{\theta^k}, \mathcal{A}))) > X(Recon(\nabla_\theta L_{\theta^k}))$, where X is a measure of defense capability (e.g. MSE) and $Recon$ is the image reconstruction.
- **(c2)** $Cost(f(\nabla_\theta L_{\theta^k}, \mathcal{A})) \leq Cost(\nabla_\theta L_{\theta^k})$, where $Cost$ is the communication cost (bits) in transmission.
- **(c3)** Allows the adjustment in the trade-off between model performance and defense capability.

We draw the attention of the reader to two important aspects of Problem 1. First, the defense capability mentioned in condition (**c1**) is a practical measure of defense success. It is the actual empirical statistic of penetration tests, different from the theoretic privacy bound of differntial privacy, which is set in advance. As a bottom-up approach, the defense capability of condition (**c1**) ensures specificity to reconstruction attacks in Problem 1, and thereof, distinctive privacy is preserved. More details are in Section 5.3.

Another point is at the location of f being at the client-side. f obscures the raw gradient into an obscured gradient that is difficult to reproduce reconstruction attacks with. Only after obscuring the gradient does a client send the information to the server. This depicts *client-owned protection* by endowing clients with their own ability to secure their own private data. In other words, clients do not have to rely on a third party for data privacy (e.g., differential privacy methods [5, 67]).

The process of differential privacy necessitates the clients to trust the server with their raw gradients. In differential privacy settings, the server inspects the clients' gradient information and alters the client gradients so that the trained model would expose minimal information on the individual clients [67]. This exposure of raw gradients is a liability allowing the server to potentially reconstruct private data. Therefore, client-owned protection is a major advantage in privacy as it restricts the accessibility of the gradient and minimizes leakage.

We concentrate on designing this obscuring function f that provides client-owned protection, and our results will aim to validate our implementation of f.

4 FRAGMENTED FEDERATED LEARNING

The feasibility of reconstruction attacks implies the significant amount of underlying information the gradient represents of the clients' private data. Our defense method Fragmented Federated Learning (FFL) designs the obscuring function f based on the abovementioned premise. Gradients are perceived as representations of

private meta-information, and the clients select and send the secure gradient layers based on the obscuring function to the central server.

In this section, we define the *global gradient*, which becomes the standard of comparison for selecting secure layers and explain the obscuring function algorithm and the total procedure of FFL.

4.1 Global Gradient

Due to the distributed foundation of FL, a model can be trained on a more general distribution of data. The aggregation of gradients (Equation 1) enhances the generalization of the training model by introducing update information on a global data collection. In this aspect, we refer to this sum $\sum_{i=1}^{N} \nabla_\theta L_{\theta^k}(x_i, y_i)$ as the *global gradient*. The global gradient reduces the bias of each client's private data on the overall model by representing the whole client data.

The central server can calculate the global gradient accurately by aggregating gradient information from all the clients. Unlike the server, each client has no access to gradients from other clients in a centralized setting. The naive action of a server providing the global gradient information to clients would naturally be to offer the aggregated global gradient to the clients along with the model parameters. But this approach entails two crucial problems:

(1) Communication cost is *doubled*.
(2) It cannot be derived from the client's allocations $\mathcal{A}$.

Any unnecessary expansion in communication cost must be avoided to be a practical FL solution. Furthermore, the global gradient does not pertain to the provided allocations of an FL client mentioned in Problem 1. Therefore, we *approximate* the global gradient in the clients' *local* environment.

The global gradient is estimated as the difference between the current model received from the central server and the previous model from supplementary storage by the Global Gradient Estimator in Figure 3. Suppose that a client receives model weights θ^k and θ^{k-1} from the server at time t_k and t_{k-1} respectively. Then,

$$\theta^k - \theta^{k-1} = \gamma \sum_{i \in C_{k,k-1}} \nabla L_\theta(x_i, y_i) \tag{4}$$

where $C_{k,k-1}$ represents the set of clients that participated in FL during time interval $[t_{k-1}, t_k]$ and γ is the learning rate. For each round of model update in FL, not all the clients are always available, and the gradients of only a sample of the total clients are used for update. Due to this random sampling of clients, if all the clients uniformly access the server and are used for model update, then the expected value of the approximation becomes:

$$\mathbf{E}[\theta^k - \theta^{k-1}] = \gamma \sum_{i=1}^{N} \nabla L_\theta(x_i, y_i) \tag{5}$$

This approximating mechanism is carried out in the Global Gradient Estimator as the difference between the parameters of the current model and the parameters of the previous model.

This simple global gradient estimation method elegantly handles both problems mentioned by the naive approach. Approximation involves only the current model weights that the client receives from the server, and the previous model weights, which were stored in the Supplementary Storage from a previous round of FL. Therefore this estimation method introduces no further communication cost and can also be derived from the allocations $\mathcal{A}$.

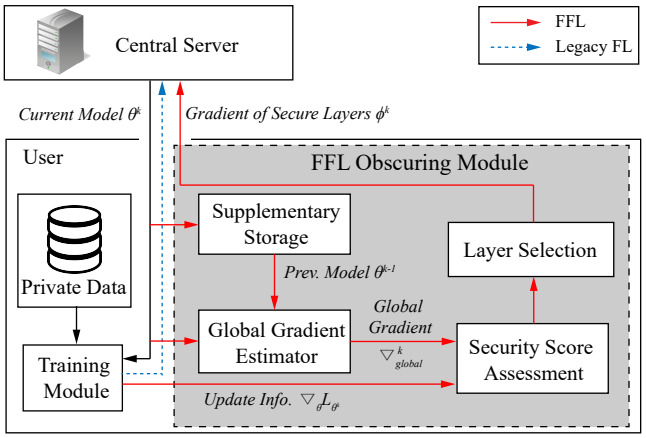

Figure 3: Workflow of FFL Framework. The blue dotted line shows the workflow of what would have been legacy FL, and the red line shows the workflow of FFL. FFL adds an Obscuring Module on to legacy FL, making it lightweight.

4.2 Gradient Obscuring in FFL

Based on the concept of global gradient, comparison with the global gradient indicates how close the client's private data distribution is with the general data distribution. Being closer to this general data distribution would imply less distinction of a client's private data, and therefore be more secure. High similarity between the layers of a client's gradient and the global gradient indicates a more generality in update information—lower vulnerability when exposed to the central server. Low similarity between the gradient and global gradient layers indicates a stronger presence of the client's private data—higher vulnerability when exposed to the central server. Based on this core foundation of selecting secure layers for distinctive privacy, we define f as such:

Definition 3 (Obscuring function f). Let $\theta^k = \{L_i^k : i \leq n\}$ be the model parameters where L_i^k denotes the parameters in the i-th layer of the model at round k and $\nabla_{global}^k = \theta^k - \theta^{k-1}$ be the global gradient. For ∇_{global}^k and input gradient $\nabla_\theta L_{\theta^k}$, the obscuring function f returns ϕ^k, the layers of the input gradient with higher cosine similarity to ∇_{global}^k where $\phi^k \subset \nabla_\theta L_{\theta^k}$. The number of layers to send to the server are predefined as a hyperparameter, layer ratio r.

Note that because we use cosine similarity as the similarity measure, γ can be ignored in Equation 5 and that the input gradient has the same shape as the model parameters (i.e. set of layers).

By employing the estimation of global gradient in FFL, the proposed obscuring function of Definition 3 becomes a solution to Problem 1. We will show through empirical evaluation that the conditions of Problem 1 are satisfied by Definition 3 in Section 6.

The visualized process of FFL is shown in Figure 3. The process of legacy FL is shown in the blue dotted line and the additions of FFL are shown in the red lines. Each client will receive the model and

Algorithm 1 Obscuring Function f of a Client in FFL

Input: Current Model Parameters θ^k, Previous Model Parameters θ^{k-1}, Maximum Number of Rounds T, Layer Ratio r, Client Data d_i, s=[]
Output: Gradient Layers $\phi^k \subset \nabla_\theta L_{\theta^k}$

while $k \leq T$ **do**
  calculate $\nabla_\theta L_{\theta^k}$ on d_i
  $\nabla^k_{global} = \theta^k - \theta^{k-1}$
    ▷ estimate Global Gradient
  for each l-th layer **do**
    `sim` $\leftarrow \langle \nabla_\theta L_{\theta^k}[l], \nabla^k_{global}[l] \rangle$
    `sim` $\leftarrow$ `sim`$/(\|\nabla_\theta L_{\theta^k}[l]\| \|\nabla^k_{global}[l]\|)$
      ▷ calculate cosine similarity
    `s` $\leftarrow$ `append( (`l`, sim) )`
  end for
  `sort(s)` in descending order of `sim` values
  truncate `s` with length $l_{trunc} = \lceil r \times$ `len(s)` $\rceil$
  $s_{trunc} = \{l \mid s[l][0], l \leq l_{trunc}\}$
  $\phi^k = \{\nabla_\theta L_{\theta^k}[i] \mid i \in s_{trunc}\}$
    ▷ obscured gradient
  $\theta^{k-1} \leftarrow \theta^k$
    ▷ update Supplementary Storage
  `return` ϕ^k and s_{trunc}
    ▷ transmit obscured gradient and indices to server
end while

train this model with their private data. The calculated gradient will then be forwarded for Security Score Assessment. In the meanwhile, the Supplementary Storage holds the previous model parameters that will be utilized for global gradient estimation, explained in Section 4.1. The Global Gradient Estimator calculates the difference between model parameters of the current model, received from this iteration, with the previous model of the Supplementary Storage.

The Security Score Assessment calculates the layer-wise similarities of the global gradient and update information by cosine similarity. These similarity values are used to rank the more secure layers of the update information. Layer Selection selects the most secure layers with least probability of exposing personal information, and are submitted to the central server. The procedure of the obscuring function f in FFL is shown in Algorithm 1.

4.3 Benefits of Layer-wise Obscuring of Gradient

The obscuring algorithm of FFL introduces additional security steps that provide protection from reconstruction attacks by evenly distributing the burden to the computational cost of individual clients.

The computations that are added by FFL for each client are *1) similarity computations* and *2) a sorting function* to rank the layer indices by security scores. The time complexity of cosine similarity is $O(n)$, where n is the number of parameters in the gradient. Sort functions are known to be $O(L \log L)$, where the L refers to the number of layers of the architecture. Unlike element-wise sorting in previous works [3, 10, 38], sorting time is decreased in magnitudes by using layers as the unit of comparison. In total, the computational cost from similarity calculations and sorting are negligible due to very low time complexities and do not stress any more computational power of the clients than when compared to general model training on private data in an FL system.

In addition to a minute addition of computational cost, FFL lessens the communication cost by decreasing the total amount of transmitted information. The server is responsible for sending the model weights to the clients, which is an inevitable procedure in terms of communication cost. For clients, the only component in transmission are the fragmented gradients. Because the ratio of gradient layer selection can be adjusted as a hyperparameter of the framework, the extent of communication cost decrease will vary by configuration.

FFL provides defense against reconstruction attacks with light computational overhead and decreased communication cost by only requiring additional storage of clients for saving the previous model.

5 IMPLEMENTATION

5.1 Simulation of Federated Learning

Standard FL setting was adopted from Liang et al. [37]. To simulate FL, several assumptions were made as follows [43]:

(1) Each client owns a non-overlapping private dataset composed from C classes.
(2) The server and all of the clients use the same architecture.
(3) A small fraction (α) of total clients (N) periodically sends queries to the server to obtain global model parameters.
(4) Equal learning opportunities are given to all the clients. In other words, each client should train a local model with the same batch size (B) and the same number of rounds (T).

Training data of each class is split into shards of equal size. Each client randomly chooses C different classes and receives a non-overlapping shard for each selected class to form a private dataset. We evaluate our FL algorithm on various settings where C is half of the total number of classes of the respective dataset.

The target architectures are conventional architectures that use convolutional layers and skip connection. We will refer to the first architecture as ConvNet, which consists of 9 convolutional layers with batch normalization (BN) [26] and rectified linear unit (ReLU) [48] activation and a single fully connected (FC) layer. The second architecture is ResNet-18 [24], which is referred to as ResNet in the evaluation for simplicity.

For the remaining assumptions, we use $\alpha = 0.1$, $B = 50$, and $N = 100$. The network is trained on 2,000 rounds for all experiments. After receiving the fragmented gradient layers, layer-wise average is conducted. The ratio of selected layers to the number of entire layers is denoted by r. We evaluate on different choices of r ($r = 0.2, 0.4, 0.6, 0.8, 1.0$) where $r = 1.0$ corresponds to standard FL.

5.2 Comparison Methods

As a defense method, FFL obscures the gradient to prevent reconstruction attacks. Due to the lack of defense works specifically targeting gradient reconstruction attacks, we compare FFL to other methods that can be thought of as defense methods against reconstruction attacks. Differential privacy is the benchmark for security

applications, as it provides theoretical bounds and allows the measurement and quantitative comparison of privacy. Although not directed at reconstruction attacks, a differentially private model provides mathematical guarantee of privacy protection against a wide range of privacy attacks; conceptually the model should defend gradient reconstruction attacks. Because the information pruning aspect of FFL could be deemed as a defense technique, we compare FFL to a gradient compression framework and attack the compressed gradient. FFL is compared with the following:

- DP [23]: A differential private FL framework from Geyer et al. We train three models by setting the total privacy budget threshold as $\epsilon = 8$ in accordance to [23], with different values of $\delta = 10^{-4}, 10^{-5}, 10^{-6}$, the probability of ϵ-differential privacy being broken. By keeping track of privacy leakage with the privacy accountant of [1], training was stopped once ϵ was reached.
- DGC [38]: Lin et al. proposes a compression method that provides communication efficiency by leveraging sparse updates. Only the highest values in the gradient are selected by compression ratio r_{comp}. Although not a defense mechanism, the concept of partial gradient selection is relevant and can be a good comparison in showing the effectiveness of global gradient in defense. We train three DGC models with compression ratio of $r_{comp} = 0.05, 0.1, 0.2$.
- FFL-random: The baseline comparison method of random selection of layers with ratio r. FFL-random represents pure fragmentation of the gradient devoid of the gradient layer selection of the obscuring function f.

5.3 Reconstruction Attack Settings

We evaluate our algorithms and baseline methods against reconstruction attack proposed by Geiping et al. [22]. We attacked FFL scenarios with [22] using the clients' gradient portions of layer ratio r. We optimize Equation 2 by replacing the gradient $\nabla_\theta L_{\theta^k}(x, y)$ with the obscured version of the gradient $f(\nabla_\theta L_{\theta^k}(x, y))$. The obscuring methods of DP (gradient clipping and perturbation) and DGC (gradient compression) also apply to this substitution. We adopt the same hyperparameter settings of [22], where the optimization runs for up to 24,000 iterations. Geiping et al. notes that this maximum iteration of 24,000 is a conservative setting and that privacy can be broken much earlier into optimization.

We use mean squared error (MSE) and peak signal-to-noise ratio (PSNR) of the images as our evaluation metric of reconstruction quality. A high MSE value denotes that the reconstructed image is dissimilar to the original image, and therefore successfully achieving defense from reconstruction attacks. A low PSNR value implies that the fidelity of the reconstructed image is corrupted by noise, and therefore successfully defending reconstruction attacks.

5.4 Datasets

CIFAR-10/100 CIFAR-10 and CIFAR-100 [35] consists of 60,000 32x32 RGB images, each of which belongs to one of 10/100 different object classes. We reconstructed 1,000 images for evaluation.

EMNIST Letters This dataset [12] consists of handwritten letter 28x28 pixel images. There are a total of 145,600 images dispersed evenly among 26 classes representing every letter of the alphabet. 1,300 images were reconstructed for evaluation.

6 FFL AS A SOLUTION

We evaluate FFL on how well the observing function f satisfies the conditions of Problem 1, in the respective order of defense capability, communication cost, and trade-off adjustment through qualitative and quantitative analysis.

6.1 Defense Capability of FFL: condition (c1)

We first look at the qualitative characteristics of reconstructed samples. Figure 4 shows reconstructed sample images by each defense method. A rather accurate reconstruction of the image is observed when using the full gradient (no FFL). Specifically for FFL, as the ratio of selected layers decreases, there is a larger degree of failure, hence, more effective defense against reconstruction attacks. The reconstructed images fail to be recognizable and display loss of visual features and definition.

Reconstruction attacks are shown to be successful for $r \geq 0.4$ when trained with FFL-random, unlike the global gradient using counterpart. This shows the unstable nature of random selection. Unlike FFL, reconstructed images of DP and DGC methods show preservation of image features, and can get a glimpse of the original image. In this aspect, FFL differs in that reconstruction of the images failed, whereas DP and DGC methods show noisy reconstructions.

Condition (**c1**) states that gradient obscuring by f should increase the defense capability. Effectiveness of FFL is shown in Figure 5 by comparison with other defense methods. The MSE and PSNR of each training setting are shown as graphs depending on the layer ratio r.

At lower layer ratio r, FFL exceeds the comparison methods in terms of defense. A larger MSE and lower PSNR value indicates higher protection against reconstruction attacks. In the case for CIFAR-10/ConvNet and CIFAR-10/ResNet, at $r = 0.4$ the MSE and PSNR values show better defense capability than DP and DGC methods of that setting (Figure 5 (a), (b)). For CIFAR-100/ResNet, the turning point for reconstruction defense is when $r = 0.4$, but the defense efficacy is sharply escalated at $r = 0.2$. At $r = 0.2$, by only paying an additional 0.6% in accuracy (Table 4), the average MSE value nearly doubles while the PSNR value plunges. In the final setting of EMNIST/ConvNet, $r = 0.2$ is the unique setting of FFL that outperforms all methods in MSE and PSNR values. For this setting, the accuracy was preserved throughout all values of r; $r = 0.2$ being the preferable setting that satisfies both accuracy and data privacy.

Interestingly enough, differential privacy shows inconsistent defense capability; the best performing trial is different for all settings. This reflects the irrelevance of differential privacy to distinctive privacy. The defense capability that differential privacy shows is not due to the privacy guarantee of the system, but due to the addition of random noise, explaining this inconsistency.

6.2 Communication Cost: condition (c2)

Obscuring function f employs layer selection, reducing the total amount of bits in transmission. This can be seen in Table 2. The following shows the average number of parameters for each round of learning with r. Because each layers in the architecture have different numbers of parameters, r does not necessarily equate to the ratio of parameters. Nevertheless, a lower ratio implies less

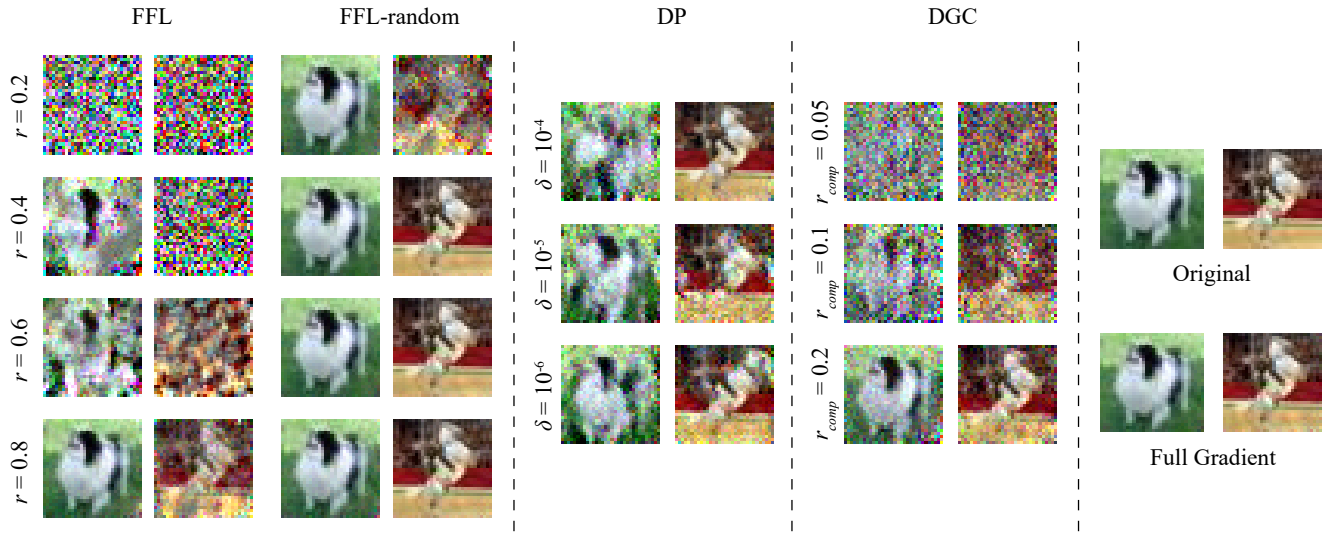

Figure 4: Reconstruction results for architecture ResNet on CIFAR-10. Each column pairs show the reconstructed results of the sample images depending on the specified method. The original image and reconstructed image using the full gradient (no FFL) are shown on the rightmost column. The respective MSE and PSNR values of each image are in Appendix A Table 6.

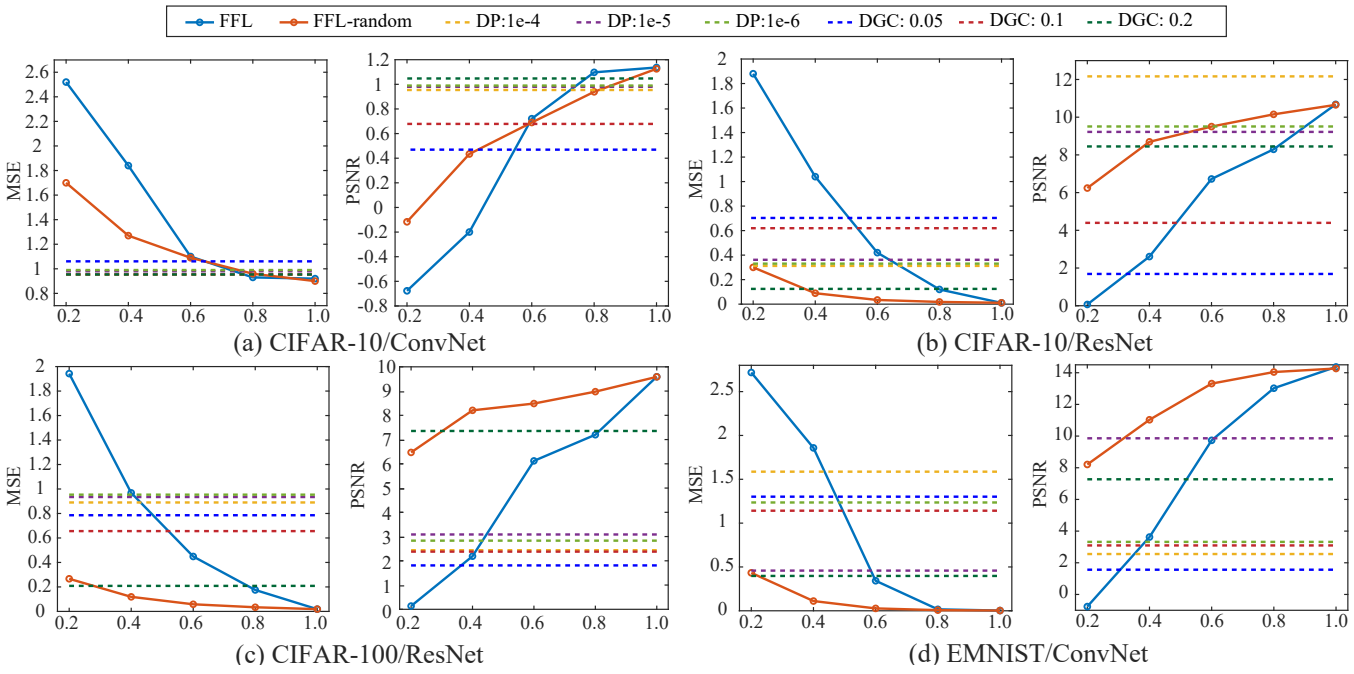

Figure 5: Quantitative evaluation of FFL on various datasets and architectures. The x-axis is the layer ratio r. DP methods are shown as constant lines because they are invariant of layer ratio r. For all settings, lower r values of FFL demonstrate greater defense capability than the comparison methods.

transmission bits, for example with $r = 0.6$ in ResNet there is an expected reduction to nearly 34% of the original parameters. f satisfies condition **(c2)** of not increasing communication cost. Additional computations are required to evaluate security scores of all the layers. Theoretically at degrees of $O(n)$ for score calculation by layer and $O(L \log L)$ for sorting by score value, we calculated the exact time ratios between training a local model and the combination of score calculation and sorting.

The computation times of a single round of FFL in each of the settings described earlier is shown in Table 3. The similarity and sorting operations of the obscuring algorithm of FFL only introduce an average of an additional 4.86% time increase. Computing the

Table 2: Transmission bits in FFL. As the layer ratio decreases, the number of parameters and therefore bits in transmission decrease, lowering communication cost.

Architecture	Layer Ratio	# of Parameters	Size
ConvNet	0.2	134K	533KB
	0.4	563K	2.24MB
	0.6	2.44M	9.72MB
	0.8	3.48M	13.8MB
	1.0	3.49M	13.9MB
ResNet	0.2	1.67M	6.69MB
	0.4	3.08M	12.3MB
	0.6	15.1M	60.5MB
	0.8	25.9M	104MB
	1.0	44.7M	179MB

Table 3: Computation time consumed in one round of FFL for training and secure layer selection. For all dataset/architecture pairs, layer selection introduces a marginal computation overhead compared to training.

Dataset/ Architecture	FFL	
	Train (sec)	Selection (sec)
CIFAR-10/ConvNet	0.94	0.03 (3.19%)
CIFAR-10/ResNet	1.27	0.09 (7.09%)
CIFAR-100/ResNet	1.34	0.10 (7.46%)
EMNIST/ConvNet	1.79	0.03 (1.68%)

security scores of each layer and selecting them according to r can be deemed negligible when compared to model training. Overall, f satisfies condition **(c2)** of not increasing communication cost.

6.3 Learning Performance of FFL: condition (c3)

Condition (**c3**) states that trade-off adjustability should be available, but this is under the premise that the FFL is *feasible*. Feasibility of FFL is shown by tolerable model accuracy. Table 4 and 5 report the accuracy of FFL on each dataset/architecture, along with the baseline of $r = 1$ and comparison models. For each training setting, the expected phenomenon of decreasing performance by r is observed. Specifically, the maximum accuracy loss per setting when compared to the baseline are 0.66, 11.34, 9.51, and 0.36 respectively. Accuracy loss is different among datasets, and judging by the outcome of the two models of CIFAR-10, also depend on what architecture is used for the dataset. Considering that accuracy itself is highly dependent on architecture, accuracy loss also being dependant is intuitive.

In the case of EMNIST/ConvNet, the accuracy stays almost constant for all ratios r. This constant performance of FFL depending on r is comparable to all methods in that setting, and can be explained by the relatively smaller complexity of the EMNIST dataset. As can be denoted by the high accuracy of EMNIST, learning EMNIST is a relatively easier task when compared with the other datasets, resulting in $r = 0.2$ to be sufficient for proper learning.

For most trials, FFL shows to be inferior in ratio-wise accuracy when compared to FFL-random. This can be understood as the cost of distinctive privacy, explained previously in Section 6.1. The selected layers of FFL are determined in the aspect of data privacy, suggesting that more private layers are in fact more valuable information when it comes to model learning.

Table 4: Accuracy on different dataset/architectures. The case of $r = 1.0$ for FFL implies the standard FedAvg [43] algorithm without using fragmented gradients. DP methods involve no fragmentation.

Dataset/ Architecture	Methods	Accuracy (%)				
		Layer Selection Ratio r				
		0.2	0.4	0.6	0.8	1.0
CIFAR-10/ ConvNet	FFL	84.42	84.28	84.31	85.05	85.08
	FFL-random	83.35	84.02	84.2	84.67	
CIFAR-10/ ResNet	FFL	78.19	80.77	86.45	89.75	89.53
	FFL-random	83.18	86.31	87.78	88.21	
CIFAR-100/ ResNet	FFL	63.48	64.02	68.94	72.29	72.99
	FFL-random	67.3	69.77	71	71.89	
EMNIST/ ConvNet	FFL	94.79	94.59	94.95	94.99	94.95
	FFL-random	94.73	94.91	94.94	95	

Table 5: The accuracy of comparison methods on different dataset/architectures with FFL at $r = 0.6$

Dataset/ Architecture	Accuracy (%)						
	DP			DGC			FFL
	10^{-4}	10^{-5}	10^{-6}	0.05	0.1	0.2	$r = 0.6$
CIFAR-10/ ConvNet	76.44	79.62	80.16	84.13	85.3	86.07	84.31
CIFAR-10/ ResNet	88.59	89.53	89.96	87.35	88.21	89.3	86.45
CIFAR-100/ ResNet	69.09	74.39	72.14	68.17	69.96	69.87	68.94
EMNIST/ ConvNet	94.69	95.01	94.95	94.95	94.98	94.87	94.95

FFL satisfies the three conditions of Problem 1. Compared with DP and DGC, it shows improved defense capability against reconstruction attacks. By being lightweight, FFL is an affordable method of providing distinctive privacy.

7 CONCLUSION

In this work, we conducted a holistic study of privacy attacks in FL and found reconstruction attacks to be unique in the form of privacy they breach. We name this privacy distinctive privacy and find that we require a light solution that can enforce distinctive privacy. Empirically, we show that FFL is a solution to this problem by satisfying the three conditions of Problem 1. Our method is practical by introducing near-negligible computation overhead without increasing communication cost when compared to legacy FL. We compare the defensive capabilities to differential privacy and DGC and show that FFL outperforms others in defense capability. We hope that our decomposition of privacy in FL can be used in the specialization of privacy-preserving methods.

ACKNOWLEDGMENTS

This research was supported by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921).

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[2] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. 2018. cpSGD: Communication-efficient and differentially-private distributed SGD. *Advances in Neural Information Processing Systems* 31 (2018).

[3] Alham Fikri Aji and Kenneth Heafield. 2017. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021* (2017).

[4] Pathum Chamikara Mahawaga Arachchige, Peter Bertok, Ibrahim Khalil, Dongxi Liu, Seyit Camtepe, and Mohammed Atiquzzaman. 2019. Local differential privacy for deep learning. *IEEE Internet of Things Journal* 7, 7 (2019), 5827–5842.

[5] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1175–1191.

[6] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210* (2018).

[7] Emmanuel J Candès, Justin Romberg, and Terence Tao. 2006. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory* 52, 2 (2006), 489–509.

[8] Qi Chai and Guang Gong. 2012. Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers. In *2012 IEEE International Conference on Communications (ICC)*. IEEE, 917–922.

[9] Sylvain Chatel, Apostolos Pyrgelis, Juan Ramón Troncoso-Pastoriza, and Jean-Pierre Hubaux. 2021. Privacy and Integrity Preserving Computations with {CRISP}. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*.

[10] Chia-Yu Chen, Jungwook Choi, Daniel Brand, Ankur Agrawal, Wei Zhang, and Kailash Gopalakrishnan. 2018. Adacomp: Adaptive residual gradient compression for data-parallel distributed training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[11] Maximin Coavoux, Shashi Narayan, and Shay B Cohen. 2018. Privacy-preserving Neural Representations of Text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 1–10.

[12] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. 2017. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2921–2926.

[13] Emiliano De Cristofaro. 2020. An overview of privacy in machine learning. *arXiv preprint arXiv:2005.08679* (2020).

[14] David L Donoho. 2006. Compressed sensing. *IEEE Transactions on information theory* 52, 4 (2006), 1289–1306.

[15] John C Duchi, Michael I Jordan, and Martin J Wainwright. 2013. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 429–438.

[16] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3-4 (2014), 211–407.

[17] Yanai Elazar and Yoav Goldberg. 2018. Adversarial Removal of Demographic Attributes from Text Data. In *EMNLP*.

[18] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 1054–1067.

[19] Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Helen Möllering, Thien Duc Nguyen, Phillip Rieger, Ahmad-Reza Sadeghi, Thomas Schneider, Hossein Yalame, et al. 2021. SAFELearn: Secure aggregation for private FEderated learning. In *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE, 56–62.

[20] Sébastien Gambs, Ahmed Gmati, and Michel Hurfin. 2012. Reconstruction attack through classifier analysis. In *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 274–281.

[21] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. 2018. 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*. 619–633.

[22] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting Gradients–How easy is it to break privacy in federated learning?. In *Advances in Neural Information Processing Systems*.

[23] Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* (2017).

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[25] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 603–618.

[26] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (Lille, France) *(ICML'15)*. JMLR.org, 448–456.

[27] Swanand Kadhe, Nived Rajaraman, O Ozan Koyluoglu, and Kannan Ramchandran. 2020. FastSecAgg: Scalable Secure Aggregation for Privacy-Preserving Federated Learning. *arXiv preprint arXiv:2009.11248* (2020).

[28] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. 2020. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence* (2020), 1–7.

[29] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM J. Comput.* 40, 3 (2011), 793–826.

[30] Deep Kawa, Sunaina Punyani, Priya Nayak, Arpita Karkera, and Varshapriya Jyotinagar. 2019. Credit Risk Assessment from Combined Bank Records using Federated Learning. (2019).

[31] Raouf Kerkouche, Gergely Ács, Claude Castelluccia, and Pierre Genevès. 2021. Compression boosts differentially private federated learning. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 304–318.

[32] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[33] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527* (2016).

[34] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).

[35] A. Krizhevsky and G. Hinton. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto* (2009).

[36] Klas Leino and Matt Fredrikson. 2020. Stolen Memories: Leveraging Model Memorization for Calibrated {White-Box} Membership Inference. In *29th USENIX Security Symposium (USENIX Security 20)*. 1605–1622.

[37] Paul Pu Liang, Terrance Liu, Ziyin Liu, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2019. Think Locally, Act Globally: Federated Learning with Local and Global Representations. *ArXiv* abs/2001.01523 (2019).

[38] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. 2018. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In *International Conference on Learning Representations*.

[39] Dong C Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming* 45, 1-3 (1989), 503–528.

[40] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. 2021. ML-Doctor: Holistic risk assessment of inference attacks against machine learning models. *arXiv preprint arXiv:2102.02551* (2021).

[41] WANG Luping, WANG Wei, and LI Bo. 2019. Cmfl: Mitigating communication overhead for federated learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 954–964.

[42] Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5188–5196.

[43] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[44] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 691–706.

[45] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 263–275.

[46] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. 2021. PPFL: privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 94–108.

[47] Daniel Moyer, Shuyang Gao, Rob Brekelmans, Aram Galstyan, and Greg Ver Steeg. 2018. Invariant representations without adversarial training. *Advances in Neural Information Processing Systems* 31 (2018).

[48] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning* (Haifa, Israel) *(ICML'10)*. Omnipress, Madison, WI, USA, 807–814.

[49] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. 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, 739–753.
[50] AJ Paverd, Andrew Martin, and Ian Brown. 2014. Modelling and automatically analysing privacy properties for honest-but-curious adversaries. *Tech. Rep* (2014).
[51] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. 2020. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*. 2021–2031.
[52] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. 2015. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, Vol. 1. IEEE, 57–64.
[53] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* (2018).
[54] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems* (2019).
[55] Mohamed Seif, Ravi Tandon, and Ming Li. 2020. Wireless federated learning with local differential privacy. In *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2604–2609.
[56] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.
[57] Jinhyun So, Basak Guler, and A Salman Avestimehr. 2020. Turbo-Aggregate: Breaking the Quadratic Aggregation Barrier in Secure Federated Learning. *arXiv preprint arXiv:2002.04156* (2020).
[58] Congzheng Song and Vitaly Shmatikov. 2020. Overlearning Reveals Sensitive Attributes. In *8th International Conference on Learning Representations, ICLR 2020*.
[59] Liwei Song and Prateek Mittal. 2021. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2615–2632.
[60] Aleksei Triastcyn and Boi Faltings. 2019. Federated learning with bayesian differential privacy. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2587–2596.
[61] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*. 1–11.
[62] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. 2020. LDP-Fed: Federated learning with local differential privacy. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*. 61–66.
[63] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440* (2020).
[64] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2512–2520.
[65] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469.
[66] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, and Heiko Ludwig. 2019. Hybridalpha: An efficient approach for privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*. 13–23.
[67] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.
[68] Wensi Yang, Yuhang Zhang, Kejiang Ye, Li Li, and Cheng-Zhong Xu. 2019. FFD: A Federated Learning Based Method for Credit Card Fraud Detection. In *International Conference on Big Data*. Springer, 18–32.
[69] Xin Yao, Tianchi Huang, Chenglei Wu, Rui-Xiao Zhang, and Lifeng Sun. 2019. Federated Learning with Additional Mechanisms on Clients to Reduce Communication Costs. *arXiv preprint arXiv:1908.05891* (2019).
[70] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 268–282.
[71] Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. 2020. Local differential privacy based federated learning for Internet of Things. *IEEE Internet of Things Journal* (2020).
[72] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*. 14774–14784.

A APPENDIX

A.1 Additional Qualitative Results

Figures 6 and 7 show additional qualitative results of our evaluation. In CIFAR-100, FFL with ResNet becomes effective since $r = 0.4$, and is the best defensive option in the case of EMNIST. In particular for the previous case, however, DGC at $r_{comp} = 0.05$ seems to show greatest defense capability with a smudging effect, whereas FFL shows to be intense noise. For EMNIST, FFL shows to be the unique defense method that properly negates reconstruction efforts. In the architecture for ConvNet for CIFAR-10, FFL-random shows unreliable results ($r = 0.6$ shows to be effective, while $r = 0.4$ is not). FFL is effective starting from $r = 0.6$, as can be seen by high distortion of the reconstructed images while the other defense options remain recognizable. Table 6 shows the specific MSE and PSNR values of the images in Figure 4.

		Dog		Horse	
Method	Variant	MSE	PSNR	MSE	PSNR
FFL	$r = 0.2$	2.5215	-4.01	2.7064	-4.32
	$r = 0.4$	1.1393	-0.57	2.5336	-4.04
	$r = 0.6$	1.3935	-1.44	1.4402	-1.58
	$r = 0.8$	0.0383	14.17	0.3009	5.22
	$r = 1.0$	0.0038	24.2	0.0097	20.12
FFL-random	$r = 0.2$	0.0234	16.32	0.7158	1.45
	$r = 0.4$	0.0019	27.14	0.0027	25.73
	$r = 0.6$	0.0054	22.71	0.0037	24.37
	$r = 0.8$	0.0185	17.32	0.0051	22.89
	$r = 1.0$	0.0038	24.2	0.0038	24.2
DP	$\delta = 10^{-4}$	1.2534	-0.98	0.0329	14.82
	$\delta = 10^{-5}$	0.5641	2.49	0.4814	3.18
	$\delta = 10^{-6}$	1.5161	-1.81	0.5078	2.94
DPC	$r_{comp} = 0.05$	1.3035	-1.15	1.2741	-1.05
	$r_{comp} = 0.1$	1.0251	-0.11	0.6485	1.88
	$r_{comp} = 0.2$	0.2506	6.01	0.1511	8.21

Table 6: MSE and PSNR values of reconstructed images in Figure 4.

A.2 Related Work

A.2.1 Server-Side Sample Reconstruction Attack and Defense. Malicious servers can exploit gradient information from clients to infer sensitive information of clients, for example, private samples of clients can be reconstructed in server. Zhu et al. [72] is the first to propose that input image can be reconstructed from gradient of its loss function with respect to model weight. Specifically, randomly initialized noisy images are optimized such that the model weight gradients and the gradient from a given sample are in close proximity in L2 distance. Wang et al. [64] applied generative adversarial network (GAN) in the gradient matching attacks with L2 distance in the scenario of FL. The recent work by Geiping et al. [22] claims that using cosine similarity as the gradient similarity metric provides higher quality of reconstructed samples than when using L2 distance, suggesting the cosine similarity reconstruction as the more sophisticated form of attack. To the best of our knowledge, we

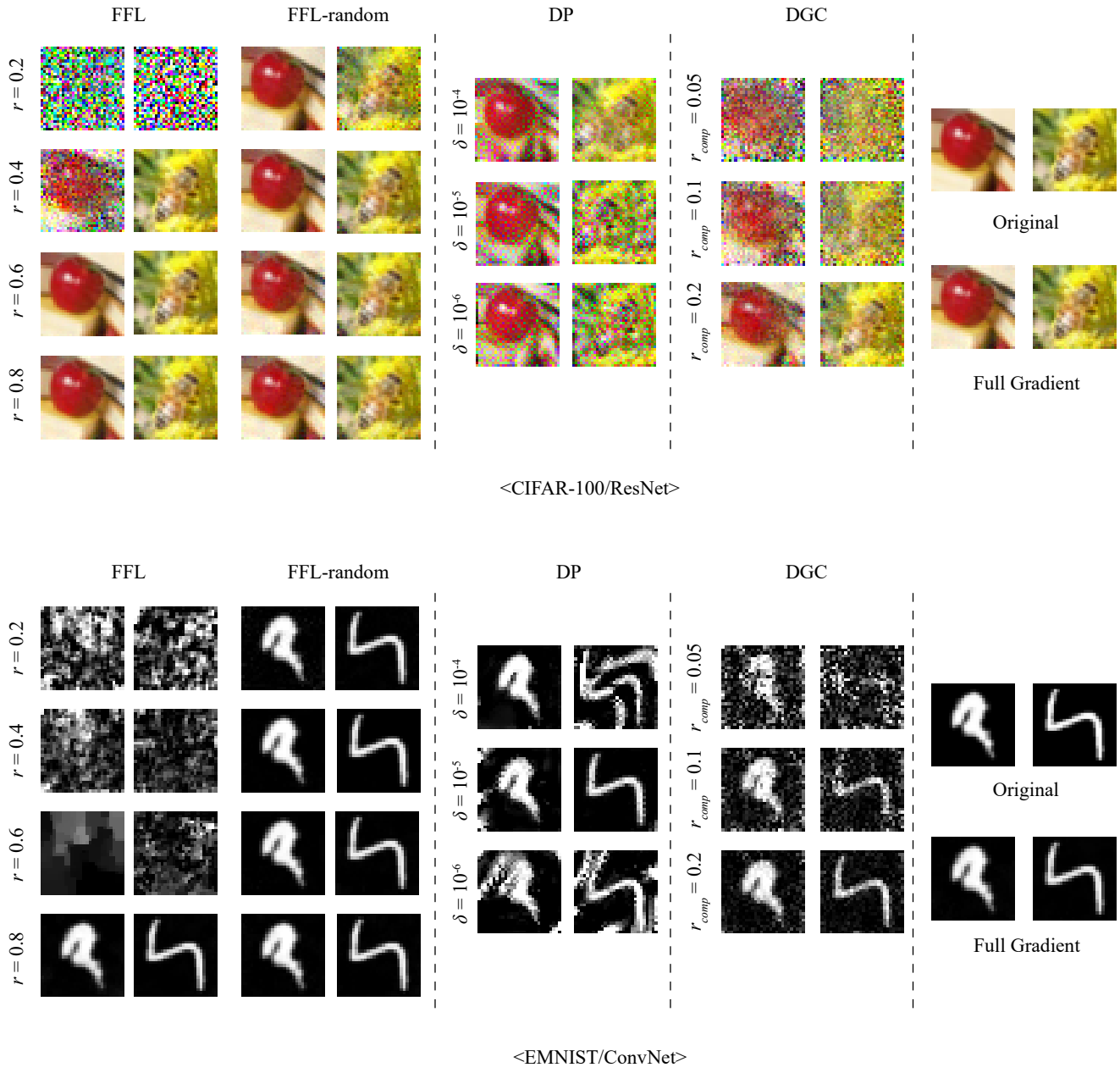

Figure 6: Reconstruction results on datasets CIFAR-100 and EMNIST. Each column pairs show the reconstructed results of the sample images depending on the specified method. The original image and reconstructed image using the full gradient (no FFL) are shown on the rightmost column.

are the first to propose a practical defense mechanism against these sample reconstruction attacks, including the attack using cosine similarity.

A.2.2 Differential Privacy. As described in Section 2.3, differential privacy is a theoretical approach to quantifying information leakage. At first, differential privacy was used in legacy (non-distributed) machine learning settings, where they proposed a stochastic gradient descent method with a moments accountant to permit tighter privacy bounds [1]. In effect, the purpose of this work was to protect a single data point's contribution. A form of differential privacy

Figure 7: Reconstruction results for architectures ConvNet on CIFAR-10. Each column pairs show the reconstructed results of the sample images depending on the specified method. The original image and reconstructed image using the full gradient (no FFL) are shown on the rightmost column.

usage in a distributed setting incorporated the Gaussian mechanism to empirically show "client level" differential privacy [23], where they aimed to protect not a single data point's contribution, but the whole client's contribution.

Other works diverged from the privacy aspect and applied differential privacy to other tasks. Agarwal et al. studies the binomial mechanism and its use in conjunction with stochastic k-level quantization for the purpose of communication efficiency [2], and Hu et al. applies differential privacy to multitask learning in federated settings by using the Gaussian mechanism.

Due to the strict conditions of (ϵ, δ) differential privacy, other works focus on ways to relax differential privacy to match their specific task. Bayesian differential privacy was proposed to relax differential privacy to obtain tighter privacy guarantees [60], and Rényi differential privacy was proposed as a natural relaxation of differential privacy based on the Rényi divergence that better suited composition of heterogeneous mechanisms [45]. Despite the multitude of differential privacy works, we are the first to test differential privacy against reconstruction attacks, and to propose that its ineffectiveness is due to it being a form of disclosure privacy.

A.2.3 Local Differential Privacy. The main problem of differential privacy was that it lacked a client-owned defense mechanism (Section 3.1). In contrast to differential privacy, local differential privacy (LDP) [15, 29] is a field where each client applies more restrictions (e.g., noise) specific to their original data (and thus achieving privacy from the server).

RAPPOR [18], the first large-scale deployment of local differential privacy, uses a memoization technique that hinders privacy attacks but allows an inevitable utility loss. Zhao et al. [71] proposes a local differential privacy algorithm specific for an Internet of Vehicles setting, and Seif et al. [55] studies FL in a Gaussian multiple access channel with LDP constraints. The above studies were tested on non-deep learning architectures, and therefore not applicable for our problem domain.

Some studies incorporated deep learning architectures [4, 62] but were limited to very shallow networks (e.g. two convolution layers). More recently, FL-CS-DP [31] proposed an LDP scheme in FL that employed compressive sensing [7, 14] to reduce communication costs, but tested on a simple architecture of a fully connected neural network with two hidden layers. Unfortunately, more complex architectures are the norm for real world applications. To reflect on this matter, FFL deals with architectures of much more depth and complexity that are inaccessible by LDP.