Enhancing Privacy Preservation in Federated Learning via Learning Rate Perturbation

Guangnian Wan[1] Haitao Du[2] Xuejing Yuan[1] Jun Yang[1] Meiling Chen[2] Jie Xu[1*]
[1] Beijing University of Posts and Telecommunications [2] China Mobile Research Institute

Abstract

Federated learning (FL) is a privacy-enhanced distributed machine learning framework, in which multiple clients collaboratively train a global model by exchanging their model updates without sharing local private data. However, the adversary can use gradient inversion attacks to reveal the clients' privacy from the shared model updates. Previous attacks assume the adversary can infer the local learning rate of each client, while we observe that: (1) using the uniformly distributed random local learning rates does not incur much accuracy loss of the global model, and (2) personalizing local learning rates can mitigate the drift issue which is caused by non-IID (identically and independently distributed) data. Moreover, we theoretically derive a convergence guarantee to FedAvg with uniformly perturbed local learning rates. Therefore, by perturbing the learning rate of each client with random noise, we propose a learning rate perturbation (LRP) defense against gradient inversion attacks. Specifically, for classification tasks, we adapt LPR to ada-LPR by personalizing the expectation of each local learning rate. The experiments show that our defenses can well enhance privacy preservation against existing gradient inversion attacks, and LRP outperforms 5 baseline defenses against a state-of-the-art gradient inversion attack. In addition, our defenses only incur minor accuracy reductions (less than 0.5%) of the global model. So they are effective in real applications.

1. Introduction

Federated learning (FL) [16, 19, 20, 33] is popularly used to meet the needs of learning from distributed data and protecting the privacy of data owners. Instead of transferring the local data, each FL client trains a model on its local data and exchanges its model update under the coordination of a central parameter server. FL leaves the training data distributed among its clients, which makes it align well with data privacy regulations, e.g., General Data Protection Regulation (GDPR) [36]. Thus, FL is suitable for developing privacy-sensitive machine learning applications such as medical services [3, 5], financial fraud detection [37], and various applications of the future sixth-generation (6G) wireless communication network [24, 31, 38].

Recent works have shown that clients' private training data may be leaked through this update-sharing scheme by gradient inversion attacks [9, 10, 14, 28, 32, 43]. Several defensive strategies have been proposed to strengthen the privacy properties of the FL system, such as differential privacy [1], secure multi-party computation [2, 39], gradient compression [29], and data representation perturbation [35]. Nonetheless, it has been demonstrated that these defenses are insufficient to provide privacy guarantees against gradient inversion attacks [28] or incur significant computational overheads [35]. Since privacy protection is the major motivation of FL, it is urgent to develop effective defenses to tackle the data leakage issue.

Assumptions of previous gradient inversion attacks. Most gradient inversion attacks assume the adversary is a FL server that is interested in unveiling the private training data of the clients from the model updates uploaded by the clients [14, 28, 32, 35], while the server must follow the FL protocol honestly and cannot modify the model architecture. Moreover, the attackers always explicitly or implicitly assume they can get the gradients of each client while the client only shares its model updates. In other words, the attackers assume they know the learning rate (LR) of each client such that they can generate the training gradients from the uploaded model updates. This assumption is realistic with existing FL algorithms since the learning rates of clients follow a unified regularity developed by the server, while the necessity of avoiding the heterogeneity of local learning rates has not been thoroughly investigated.

A natural question is: *Can clients utilize local learning rates to protect their data?* To investigate this, we form a FL system composed of two clients with non-IID data from the MNIST dataset and adjust the clients' learning rates to train the models. Compared with allocating a unified learning rate by the server, we find randomizing the local learning rates only causes small fluctuations (about 0.2%) in

*Corresponding author: *Jie Xu (cheer1107@bupt.edu.cn).*

the global models' accuracy, and allocating relatively bigger learning rate expectations to clients with higher-quality datasets can suppress FL's drift [17] issue introduced by non-IID data and thus improve the model's accuracy.

Based on our above observations, we propose LRP, a novel defense that perturbs every client's learning rates, such that the learning rates appear uniformly random. For classification tasks on non-IID data, we present an adaptive defense (ada-LRP) to improve the model's accuracy by personalizing the expectations of local learning rates. As the server cannot extract clients' exact gradients without knowing their learning rates, the data reconstructed by gradient inversion attacks can be significantly degraded. In addition, we derive a convergence guarantee to FedAvg with perturbed local learning rates on non-IID data. The image classification experiments on MNIST [23], CIFAR-10 [21], CIFAR100 [21], and ImageNet [6] show that our defenses do not incur much accuracy loss (less than 0.5%). We conduct experiments on MNIST, CIFAR-100, and LFW [13] for defending against the Deep Leakage from Gradient (DLG) [43] attack and Improved DLG (iDLG) [42] attack, and on ImageNet against Generative Gradient Leakage (GGL) [28] attack. The results show that our defenses successfully enhance the privacy preservation of FL against gradient inversion attacks.

Our main contributions are summarized as follows:

- *Findings.* By analyzing the impact of randomizing and personalizing local learning rates on FL, we find that (1) setting the local learning rates to be uniformly random has a minor effect on test accuracy but significantly degrades gradient inversion attacks, (2) FedAvg with uniformly distributed random learning rates converges well for strongly convex and smooth problems, and (3) scaling the local learning rates to be personalized values can mitigate the drift issue suffered by FL and thus improve the accuracy of the global model.
- *Effective defenses against gradient inversion attack.* We propose a learning-rate-perturbation-based defense (LRP), which outperforms five existing defenses on five metrics (e.g., MSE-R and LPIPS) against a state-of-the-art gradient inversion attack. Besides, for classification tasks, our adaptive defense ada-LRP improves the global model's accuracy compared with LRP.

2. Related Work

2.1. Gradient Inversion Attacks in FL

Gradient inversion attacks originate from DLG proposed by Zhu *et al.* [43], which reconstructs clients' private data by minimizing the distance between gradients from generated dummy data and real data. On top of this work, iDLG [42] analytically extracts the labels from the gradients to improve the attack. As follow-up works, Geiping *et al.* [10] succeed in reconstructing ImageNet-level resolution data samples, and yin *et al.* [40] improve the attack by introducing the batch normalization priors. However, the FL clients may not share their private batch normalization statistics [14, 27]. An orthogonal line of work by Fowl *et al.* [9] introduces a new analytic attack to reconstruct clients' data samples while the server is malicious. Lu *et al.* [32] propose APRIL to recover clients' data from gradients of self-attention-based models. Additionally, Li *et al.* propose GGL [28], which substantially enhances the attack by leveraging generative adversarial networks (GAN), and the work validates that clients' private data can still be reconstructed under several privacy-preserving settings.

2.2. Privacy Preservation in FL

Existing defense methods for privacy preservation in FL can be generally categorized into two types: encrypting updates and perturbing updates. Cryptographic approaches prevent data leakage by encrypting updates to achieve multi-party computation (MPC) [2, 12]. However, these approaches can incur unignorable computational overheads, and merely relying on MPC cannot provide sufficient privacy guarantees to preserve clients' privacy [9, 28, 35]. Perturbing updates is another line of research, which perturbs the updates shared by clients to degrade the information inferred by adversaries.

In detail, differential privacy (DP) [1, 16] is a straightforward method to perturb the clients' shared updates through clipping and adding noise to gradients. Nevertheless, for preventing gradient inversion attacks, DP requires adding too much noise which can cause unneglectable accuracy loss [43]. In addition, gradient compression [29] is effective to degrade the gradient inversion attack. More recently, Sun *et al.* [35] propose a new privacy-preserving defense named Soteria, which perturbs the data representation such that the reconstructed data by adversaries is dissimilar to the clients' raw data. However, merely applying these methods is demonstrated to be not sufficient to prevent information leakage against state-of-the-art attacks [28].

3. Preliminaries

3.1. Federated Averaging (FedAvg)

The recent multitude of federated learning algorithms can be understood as variants of FedAvg [33]. Thus, it is natural that we start from FedAvg. In classical FedAvg, the objective functions of the FL system are defined as:

$$\min_{W} \left\{ \mathcal{L}(W, \mathcal{D}_{global}) \triangleq \sum_{k=1}^{N} p_k \mathcal{L}_k(W, \mathcal{D}_k) \right\} \quad (1)$$

$$\mathcal{L}(W, \mathcal{D}) \triangleq \frac{1}{|\mathcal{D}|} \sum\nolimits_{x \in \mathcal{D}} \ell(W; x),$$

where N is the numer of clients, p_k is the aggregation weight of the $k-th$ client, $p_k \geq 0$ and $\sum_{k=1}^{N} p_k = 1$, $\ell(\cdot;\cdot)$ is a user-specified loss function, and $\mathcal{D}$ is the dataset for training.

After receiving the model parameters W_t from the central server, every client (i.e., $k-th$) lets $W_t^k = W_t$ and performs E (≥ 1) local steps [1]

$$W_{t+i+1}^k \longleftarrow W_{t+i}^k - \eta_{t+i} \nabla \mathcal{L}_k(W_{t+i}^k, \xi_{t+i}^k), \quad (2)$$

where η_{t+i} denotes the learning rate and ξ_{t+i}^k is a training batch sampled from the local data. Then each client sends W_{t+E} to the server.

In a realistic FL system, the server can only collect the outputs of the first K responded clients. Supposing the first K responded clients form a set S_t ($|S_t| = K$), the server aggregates the clients model parameters following

$$W_{t+E} \longleftarrow \frac{N}{K} \sum_{k \in S_t} p_k W_{t+E}^k. \quad (3)$$

3.2. Gradient Inversion Attack

After the local training stage, the curious server can collect the clients' model updates and extract a client's gradients g computed on its local data $\{x, y\}$. Given the global model parameters W and the client's gradients g, the server could reveal the client's private local data by generating an $\{x^*, y^*\}$ with an objective function:

$$\underset{(x^*,\, y^*)}{argmin}\ \mathcal{L}_{grad}(x^*, y^*, W, g) + \alpha \mathcal{P}(x^*), \quad (4)$$

where $\mathcal{L}_{grad}(x^*, y^*, W, g)$ denotes the matching loss of the gradients generated from reconstructed data $\{x^*, y^*\}$ with the provided real gradients g. The attacker may leverage GAN to generate $\{x^*, y^*\}$ [28]. $\mathcal{P}(x^*)$ is a regularization term based on the adversary's prior knowledge [14].

4. FL with Learning Rate Perturbation

In this section, we first show that (1) the local learning rates can be hidden from the server by adding perturbation and (2) scaling the clients' learning rates can mitigate the drift [17] issue of FL under non-IID data. We also derive a convergence guarantee of FedAvg with perturbed local learning rates. Finally, we present our defense algorithms against gradient inversion attacks.

4.1. Personalizing and Randomizing Local LRs

The non-IID data across the clients is one of the key challenges in FL, which can introduce a drift in the local and global updates, as shown in the left part of Figure 1.

[1] Unlike the original paper [33], we use E to denote the times of local steps instead of epochs, following [26].

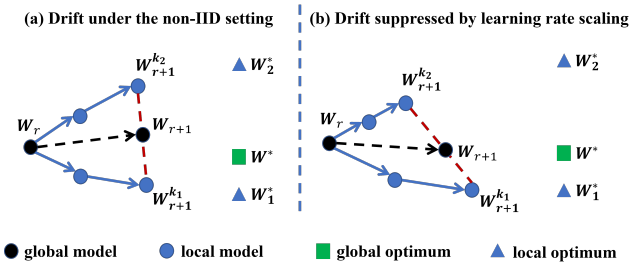


Figure 1: Examples of drift in FL on non-IID data.

As the distribution of each client's dataset is not identical to the global distribution, the local optimum of each client is inconsistent with the global optimum. Since each local model is updated towards its local optimum, the global update, which is the average of the clients' updates, may drift from the direction towards the global optimum, which can significantly influence the FL model's accuracy. A natural thought is: *The direction of the global update can be corrected by enabling clients with optima closer to the global optimum to use larger local step sizes (i.e., learning rates)*, as shown in the right part of Figure 1.

To validate our thought, we form a simple FL task with two clients where the datasets of the clients are non-IID. In detail, we first sort the training data of the MNIST dataset by label, then divide it into 600 shards of size 100. For each of the first two labels, we choose four shards and assign them to the first client. Then for each of the last eight labels, we choose one shard and assign them to the second client. Therefore, both clients do not have data of all ten digits, and the labels of their data are not overlapped. It allows us to explore what perturbing the local learning rate will introduce to FL under a non-IID setting.

We experiment on a convolutional neural network (CNN), a multilayer perceptron (MLP), and a logistic regression, respectively. The models are trained for 100 communication rounds following FedAvg, and the two clients may use different learning rates sampled from {0.005, 0.01, 0.02}. Besides, to investigate whether perturbing learning rate causes much accuracy loss, there are also sets of experimental cases where each client randomly samples their learning rates in each local training step following uniform distributions with a minimum of 0 and expectations of 0.005, 0.01, and 0.02, respectively.

As shown in Table 1, on all three models, we get the best test accuracy with $\{\eta_1, \eta_2\}$ = {0.005, 0.02} and the worst with $\{\eta_1, \eta_2\}$ = {0.02, 0.005}. This indicates that (1) personalizing clients' learning rates can mitigate or aggravate the drift issue of non-IID FL and thus impact the global model's performance. In addition, the experimental results with and without learning rate perturbation show that (2) setting the local learning rates to be uniformly random in every local training step does not cause unneglectable accu-

Table 1: Comparison of test accuracy (%) with different learning rate combinations, including the mean and standard deviation across 3 runs. The η_1 and η_2 indicate the local learning rates of the two clients. The ✓ in the column of perturbation indicates the two clients' learning rates are randomly sampled following uniform distributions with expectations η_1 and η_2.

Model	Perturbation	$\eta_1 = 0.005$			$\eta_1 = 0.01$			$\eta_1 = 0.02$		
		$\eta_2 = 0.005$	$\eta_2 = 0.01$	$\eta_2 = 0.02$	$\eta_2 = 0.005$	$\eta_2 = 0.01$	$\eta_2 = 0.02$	$\eta_2 = 0.005$	$\eta_2 = 0.01$	$\eta_2 = 0.02$
CNN		88.36 ± 0.17	92.10 ± 0.05	$\mathbf{94.21 \pm 0.04}$	87.07 ± 0.71	91.55 ± 0.25	93.96 ± 0.17	$\mathbf{85.97 \pm 0.55}$	90.98 ± 0.52	93.56 ± 0.17
	✓	87.97 ± 0.24	92.00 ± 0.08	$\mathbf{94.39 \pm 0.02}$	86.91 ± 0.31	91.32 ± 0.25	93.95 ± 0.09	$\mathbf{85.53 \pm 0.54}$	90.34 ± 0.19	93.59 ± 0.25
MLP		87.22 ± 0.07	88.62 ± 0.07	$\mathbf{89.18 \pm 0.14}$	86.44 ± 0.04	88.08 ± 0.08	88.84 ± 0.13	$\mathbf{85.50 \pm 0.05}$	87.61 ± 0.11	88.50 ± 0.19
	✓	87.18 ± 0.09	88.59 ± 0.12	$\mathbf{89.18 \pm 0.08}$	86.35 ± 0.05	88.16 ± 0.07	89.02 ± 0.05	$\mathbf{85.46 \pm 0.10}$	87.67 ± 0.11	88.63 ± 0.11
Logistic Regression		86.84 ± 0.02	87.14 ± 0.02	$\mathbf{87.22 \pm 0.01}$	86.21 ± 0.03	86.78 ± 0.02	86.83 ± 0.01	$\mathbf{85.35 \pm 0.03}$	86.14 ± 0.00	86.46 ± 0.04
	✓	86.84 ± 0.05	87.14 ± 0.04	$\mathbf{87.24 \pm 0.03}$	86.26 ± 0.03	86.78 ± 0.01	86.88 ± 0.04	$\mathbf{85.48 \pm 0.09}$	86.29 ± 0.06	86.46 ± 0.05

racy loss. Our experiments using Adam [18] for local training have similar results, which are shown in the Appendix.

4.2. Convergence Analysis

The observations in section 4.1 give an intuition that randomizing FL clients' learning rate does not incur much accuracy loss. In this section, we derive the convergence guarantee of FedAvg with uniformly random local learning rates on non-IID data. The analysis is inspired by Li *et al.* [26].

Since each client trains its local model with random and different learning rates, eq. (2) is changed to:

$$W^k_{t+i+1} \longleftarrow W^k_{t+i} - \eta^k_{t+i} \nabla \mathcal{L}_k(W^k_{t+i}, \xi^k_{t+i}), \quad (5)$$

where every client perturbs its learning rate to be values following a uniform distribution with expectation η in every local step t:

$$\eta^k_t \sim \mathcal{U}(0, 2\eta). \quad (6)$$

We make the following assumptions, which have been also made by the work [26].

Assumption 1. *$\mathcal{L}_1, \cdots, \mathcal{L}_K$ are all L-smooth: for all V and W, $\mathcal{L}_k(V) \leq \mathcal{L}_k(W) + (V-W)^T \nabla \mathcal{L}_k(W) + \frac{L}{2}\|V - W\|_2^2$.*

Assumption 2. *$\mathcal{L}_1, \cdots, \mathcal{L}_K$ are all μ-strongly convex: for all V and W, $\mathcal{L}_k(V) \geq \mathcal{L}_k(W) + (V-W)^T \nabla \mathcal{L}_k(W) + \frac{\mu}{2}\|V - W\|_2^2$.*

Assumption 3. *Let ξ^k_t be sampled from the k-th device's local data with batch size B. The variance of stochastic gradients in each device is bounded: $\mathbb{E}\left\|\nabla \mathcal{L}_k(W^k_t, \xi^k_t) - \nabla \mathcal{L}_k(W^k_t)\right\|^2 \leq \sigma_k^2$ for $k = 1, \cdots, N$.*

Assumption 4. *The expected squared norm of stochastic gradients is uniformly bounded, i.e., $\mathbb{E}\left\|\nabla \mathcal{L}_k(W^k_t, \xi^k_t)\right\|^2 \leq G^2$ for all $k = 1, \cdots, N$ and $t = 1, \cdots, T-1$*

Assumption 5. *Assume S_t contains a subset of K indices uniformly sampled from $[N]$ without replacement. Assume the data is balanced in the sense that $p_1 = \cdots = p_N = \frac{1}{N}$. The aggregation step performs $W_t \longleftarrow \frac{N}{K}\sum_{k \in S_t} p_k W^k_t$.*

Let $\mathcal{L}^*$ and $\mathcal{L}^*_k$ be the minimum values of $\mathcal{L}$ and $\mathcal{L}_k$, respectively. We use the term $\Gamma = \mathcal{L}^* - \sum_{k=1}^N p_k \mathcal{L}^*_k$ for quantifying the degree of non-IID. If the data are non-IID, then Γ is nonzero, and its magnitude reflects the heterogeneity of the data distribution.

Theorem 1. *Let Assumptions 1 to 5 hold and L, μ, σ_k, G be defined therein. Choose $\kappa = \frac{L}{\mu}$, $\gamma = \max\{8\kappa, E\}$ and the expected learning rate $\eta_t = \frac{2}{\mu(\gamma + t)}$. Then FedAvg with uniformly perturbed local learning rates satisfies*

$$\mathbb{E}\left[\mathcal{L}(W_T)\right] - \mathcal{L}^* \leq \frac{\kappa}{\gamma + T}\left(\frac{2(B+C)}{\mu} + \frac{\mu\gamma}{2}D\right), \quad (7)$$

$$B = \sum_{k=1}^N p_k^2(\sigma_k^2 + \frac{1}{3}G^2) + 6L\Gamma + 32(E-1)^2G^2,$$
$$C = \frac{N-K}{N-1}\frac{16}{K}E^2G^2, D = \mathcal{L}\|W_1 - W^*\|^2.$$

Assumption 5 requires $p_k = \frac{1}{N}$ for any client, which is unrealistic. To address this, we first transform $\widetilde{\mathcal{L}}_k(W) = p_k N \mathcal{L}_k(W)$ be a scaled local objective function. Thus

$$\mathcal{L}(W) = \sum_{k=1}^N p_k \mathcal{L}_k(W) = \frac{1}{N}\sum_{k=1}^N \widetilde{\mathcal{L}}_k(W). \quad (8)$$

Then each client performs its local updates following:

$$\begin{aligned} W^k_{t+i+1} &= W^k_{t+i} - \eta^k_{t+i} \nabla \widetilde{\mathcal{L}}_k(W^k_{t+i}, \xi^k_{t+i}) \\ &= W^k_{t+i} - \widetilde{\eta}^k_{t+i} \nabla \mathcal{L}_k(W^k_{t+i}, \xi^k_{t+i}), \end{aligned} \quad (9)$$

where $\widetilde{\eta}^k_{t+i}$ is transformed to $p_k N \eta^k_{t+i}$, which increases the heterogeneity of local learning rates. And if we replace L, μ, σ_k, G to $\widetilde{L} \triangleq \nu L$, $\widetilde{\mu} \triangleq \varsigma\mu$, $\widetilde{\sigma}_k = \sqrt{\nu}\sigma$, and $\widetilde{G} = \sqrt{\nu}G$, Theorem 1 holds. Here, $\nu = N \cdot \max_k p_k$ and $\varsigma = N \cdot \min_k p_k$. We put the proof in the Appendix.

4.3. Defense Algorithm

Based on our observations in section 4.1 and theoretical analysis in section 4.2, we propose a defense against gradient inversion attacks by perturbing every client's learning rates in every local training step. Without loss of generality, we build the algorithm based on FedAvg, which is the fundamental FL algorithm, and it can be naturally generalized to other FL algorithms.

Algorithm 1 details the training process with our defenses. After receiving the learning rate η initialized by the

Algorithm 1 FL with learning rate perturbation. We use blue color to mark the part for adaptive algorithm.

Input: The N local clients are indexed by k, local batch size B, learning rate initialized by server η, proportion of selected clients in every global communication round C, number of global communication rounds R, number of local steps E.
Output: The final model.
Server executes:

initialize W_0
compute and broadcast $\sum_{k=1}^{N} n_k$ and $\sum_{k=1}^{N} |\mathcal{Y}_k|$
for each global round r = 0, 1, 2, ... R-1 **do**
  $K \leftarrow max(C \cdot N, 1)$
  $S_t \leftarrow$ (random set of K clients)
  for each cleint $k \in S_t$ **in parallel do**
    $W_{r+1}^k \leftarrow \text{ClientUpdate}(k, W_r)$
  end for
  $W_{r+1} \leftarrow \frac{1}{K} \sum_{k=1}^{K} W_{r+1}^k$
end for
return W_R

ClientUpdate(k, W)**:**

$\mathcal{B} \leftarrow$ (split the local dataset into batches of size B)
$\widetilde{\eta} \leftarrow p_k N \eta$
$\mathcal{K} \leftarrow \zeta \cdot (|\mathcal{Y}_k| - 1/N \cdot \sum_{i=1}^{N} |\mathcal{Y}_i|) + \beta$
for each local step t = 0, 1,2,... E-1 **do**
  generate (i) $\eta_t^k \sim \mathcal{U}(0, 2\widetilde{\eta})$ or (ii) $\eta_t^k \sim \mathcal{U}(0, 2\mathcal{K}\widetilde{\eta})$
  // update local model on mini-batch $b \in \mathcal{B}$
  $W \leftarrow W - \eta_t^k \nabla \mathcal{L}(W, b)$
end for
return W

server, every client perturbs its learning rate by sampling a random value η_t^k which follows a uniform distribution with expectation η in every local step t:

$$\eta_t^k \sim \mathcal{U}(0, 2\widetilde{\eta}), \tag{10}$$

where $\mathcal{U}(0, 2\widetilde{\eta})$ denotes a uniform distribution with a minimum of 0 and a maximum of $2\widetilde{\eta}$ where $\widetilde{\eta} = p_k N \eta$.

Since the local learning rates appear uniformly random to the server, the assumption that the server knows the exact learning rate of every client is relaxed. Thus, the gradients g_r^{k*} inferred by the server from W_{r+1}^k will be different from the real gradients, which can degrade the gradient inversion attack by misleading it into wrong reconstructions:

$$\underset{(x^*,\, y^*)}{argmin}\ \mathcal{L}_{grad}(x^*, y^*, W_r^k, g_r^{k*}) \neq (x, y). \tag{11}$$

We also observe the objective function misleading phenomenon in our experiments, which is shown in Figure 4 in section 5.

The observations in section 4.1 indicate personalizing clients' local learning rates can mitigate the drift issue. Therefore, to improve the model's accuracy and further increase the range of random values, we present an adaptive method for the classification task in label distribution skew settings, by which each client samples its learning rates w.r.t. the number of classes it possesses:

$$\eta_t^k \sim \mathcal{U}\big(0, 2\mathcal{K}(|\mathcal{Y}_k|)\widetilde{\eta}\big), \tag{12}$$

where $|\mathcal{Y}_k|$ is the number of classes in the k^{th} client's dataset. Since the local optimum of clients possessing more classes of data tends to be closer to the global optimum, $\mathcal{K}(|\mathcal{Y}_k|)$ is a monotonically increasing function to allow clients with more classes of data to use relatively larger learning rate expectations. We find that a simple linear function is effective in scaling the local learning rates and thus improves the model performance:

$$\mathcal{K}(|\mathcal{Y}_k|) = \zeta \cdot (|\mathcal{Y}_k| - \frac{1}{N} \sum_{i=1}^{N} |\mathcal{Y}_i|) + \beta, \tag{13}$$

where ζ and β are two introduced parameters to control the range of random values. For instance, β can be set to 1 to keep the average expectation of all clients' learning rates to be $\widetilde{\eta}$.

As shown in Algorithm 1, unlike other FL algorithms, there is no need for any client to send the size of local datasets n_k or the local class number $|\mathcal{Y}_k|$ to the server. But each client may need the sum of them to generate p_k and $\mathcal{K}(|\mathcal{Y}_k|)$. We model this as a secure multi-party computation problem and present a brief protocol inspired by [2] when the server is honest-but-curious which faithfully delivers all messages between users and a public-key infrastructure (PKI) exists.

Each pair of clients first agree on a random value s_{k_1,k_2} using Diffie-Hellman (DH) server-mediated key agreement scheme [8]. Then each client computes

$$n_{k_1}^* = n_{k_1} + \sum_{k_1 > k_2} s_{k_1,k_2} - \sum_{k_1 < k_2} s_{k_1,k_2}, \tag{14}$$

and sends $n_{k_1}^*$ to the server. The server computes $\sum_{k_1=1}^{N} n_{k_1}^*$ which is equal to $\sum_{k_1=1}^{N} n_{k_1}$ and sends it back to every client. The sum of $|\mathcal{Y}_k|$ can be computed in the same way. If the server wishes to obtain the s_{k_1,k_2} in a DH key agreement process from the messages it receives, it has to solve a Discrete logarithm Problem that belongs to the class NP (nondeterministic, polynomial). Therefore, each client can get $\sum_{k=1}^{N} n_k$ and $\sum_{k=1}^{N} |\mathcal{Y}_k|$ while n_k and $|\mathcal{Y}_k|$ is preserved locally, which prevent the server from inferring the expectation of each client's local learning rates.

5. Experiments

5.1. Experimental Setup

FL Task and Datasets. We compare the accuracy of models trained with our algorithms and FedAvg on the

Table 2: Test accuracy (%) on MNIST, CIFAR-10, CIFAR-100, and ImageNet for vanilla training and LRP, including the mean and standard deviation of test accuracy across 3 runs except for ImageNet.

Non-IID	**Local learning rate**	**MNIST**	**CIFAR-10**	**CIFAR-100**	**ImageNet**
Vanilla training (Schme I)	$\eta_k = \eta$	98.84 ± 0.01	92.99 ± 0.19	65.69 ± 0.61	61.52(83.10)
Vanilla training (Schme II)	$\eta_k = \eta$	98.36 ± 0.05	93.01 ± 0.09	65.56 ± 0.45	
LRP	$\eta_k \sim \mathcal{U}(0, 2p_k N\eta)$	98.90 ± 0.02	93.03 ± 0.17	65.44 ± 0.36	
Ada-LRP	$\eta_k \sim \mathcal{U}(0, 2\mathcal{K}p_k N\eta)$	98.97 ± 0.04	93.32 ± 0.14	65.69 ± 0.29	61.66(83.04)
IID	**Local learning rate**	**MNIST**	**CIFAR-10**	**CIFAR-100**	
Vanilla training	$\eta_k = \eta$	99.20 ± 0.02	94.30 ± 0.09	72.15 ± 0.21	
LRP	$\eta_k \sim \mathcal{U}(0, 2\eta)$	99.19 ± 0.01	94.15 ± 0.07	72.65 ± 0.29	

MNIST, CIFAR-10, CIFAR-100, and ImageNet ILSVRC 2012 datasets for image classification tasks. For the non-IID setting, we first sort the training data by label, divide it into shards, and assign these shards randomly to 100 clients. Each shard contains examples of at most two classes and most shards only have examples of one class. The size of each shard of the MNIST dataset is 200 and each client will have at least one shard and at most nine shards. For CIFAR-10, CIFAR-100, and ImageNet, we choose 50 as the shard size. Each client can get at least one shard and at most nine shards of CIFAR-10 and CIFAR-100 datasets. And we assign each client at least 50 shards and at most 200 shards of the ImageNet dataset. Our dataset partition is similar to the configuration in [33], while in the work of McMahan *et al.* [33], each client can get an identical number of shards, which is less realistic than ours. In addition, we also construct IID datasets from MNIST, CIFAR-10, and CIFAR-100 where the training data is shuffled and uniformly partitioned into 100 clients. We reserve 20% of each client's data for validation. The random seeds for repeated trials are set to be 1024, 1022, and 1020.

Attack methods. We evaluate the effectiveness of our defenses against three gradient inversion attacks. (1) **DLG attack** [43] generates reconstructed data with the objective function to minimize the distance between the gradients uploaded by clients and the gradients generated by the reconstructed data. (2) **iDLG attack** [42] optimize reconstructed inputs with the same idea as DLG while it analytically extracts ground-truth labels from the gradients. (3) **GGL attack** [28] leverages the latent space of generative adversarial networks (GAN) as a prior, which is a state-of-the-art attack that can recover high-resolution images from the gradients under its considered defense settings. Therefore, we use this attack to compare LRP with defense baselines.

Defense baselines. We compare our proposed defense with the following existing defenses. (1) **Additive noise** [1, 28, 35, 43] injects noise into the gradients. We apply Gaussian noise with $\sigma = 0.1$ and central 0. (2) **Gradient Clipping** [1, 11] clips the gradient as $\mathcal{T}_{cli}(g, S) = g/\max(1, \frac{||g||_2}{S})$ and we set the bound S as 4. (3) **Differential privacy** [1] combines the gradient clipping and additive noise. (4) **Gradient compression** [29] prunes the gradients below a given threshold. We prune 90% of the original gradients. (5) **Soteria** [28, 35] prunes the representations embedded in the gradients with a rate of 0.8.

Hyperparameter configurations. For the image classification tasks, we set the number of local steps E as 50 for MNIST, CIFAR-10, and CIFAR-100 and 2000 for ImageNet. The batch size B is set as 32, 40, 40, and 64 for MNIST, CIFAR-10, CIFAR-100, and ImageNet, respectively. The number of sampled devices in each communication round is 10. For DLG and iDLG attacks, we apply L-BFGS [30] with learning rate 1, history size 100, max iterations 20, and optimize for 1000 iterations. For the GGL attack, following the configuration of [28], we use a pre-trained BigGAN [4] as a prior and CMA-ES as the optimizer. The number of updates is set to 800.

5.2. Accuracy Results

We report the accuracy results across 3 repeated trials in Table 2. Since the implementation of the aggregation scheme of FedAvg can vary, we choose two schemes: $W_t = \frac{N}{K}\sum_{k\in S_t} p_k W_t^k$ (Scheme I) [26] and $W_t = \sum_{k\in S_t} \frac{p_k}{\sum_{i\in S_t} p_i} W_t^k$ (Scheme II) [25], both as vanilla training algorithms. For the firsr scheme, we transform $\mathcal{L}_k$ to $p_k N\mathcal{L}_k$ to keep $p_k = 1/N$ as [26] did. Note that LRP and ada-LRP are developed based on Scheme I. For ada-LRP, we set $\beta = 1$ to keep the average expectation across all the clients to be η. And ζ are chosen to be $1/6, 1/10, 1/20, 1/400$ for MNIST, CIFAR-10, CIFAR-100, and ImageNet respectively.

We use a shallow CNN model architecture for MNIST, ResNet-18 architecture for CIFAR-10 and CIFAR-100, and RexNet-130 architecture for ImageNet. We apply data augmentation to CIFAR-10 and CIFAR-100 datasets following [7]. The initial learning rate is set to 0.01 for MNIST, CIFAR-10, and CIFAR-100 and 0.1 for ImageNet. On the four datasets, we use a cosine learning rate decay. Please refer to the Appendix for other implementation details.

As shown in Table 2, we report the top-1 test-set accuracy of the trained global model on MNIST, CIFAR-10,

Table 3: Quantitative comparison of our defense with baseline methods against GGL attack. ↑: the higher the metric the better privacy preservation. ↓: the lower the metric the better privacy preservation.

Evaluation Metric	Additive Noise	Gradient Clipping	Clipping+Noise	Gradient Compression	Soteria	LRP
MSE (↑)	0.2746	0.2335	0.3443	0.2117	0.2079	**0.4627**
PSNR (↓)	5.7325	6.4000	4.7408	5.7545	6.8540	**4.1501**
LPIPS (VGG) (↑)	0.6170	0.5808	0.6448	0.6068	0.5748	**0.6966**
LPIPS (ALEX) (↑)	0.5215	0.4755	0.5703	0.4895	0.4538	**0.6227**
MSE-R (↑)	0.0014	0.0014	0.0114	0.0017	0.0018	**0.3181**

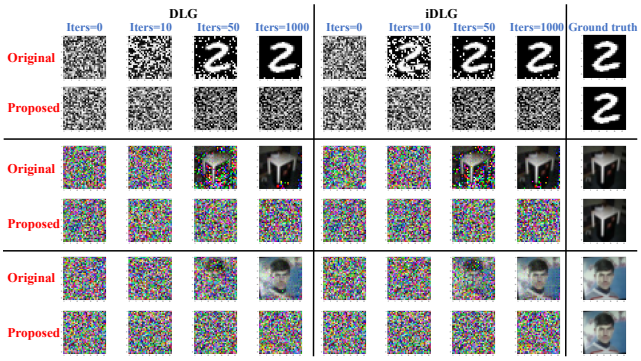


Figure 2: Defense against DLG and iDLG on images from MNIST, CIFAR-100, and LFW, respectively.

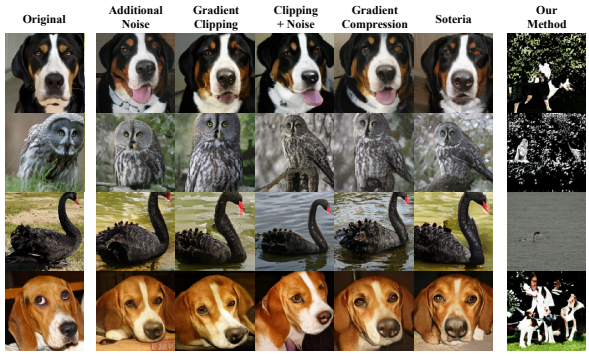


Figure 3: Visual comparison of our method with defense baselines against GGL: original images (first column) and their reconstructions (the rest of columns).

and CIFAR-100, and the top-1 and top-5 accuracy on the 50k validation images of ImageNet, including the mean and standard deviation across 3 runs with different random seeds except for ImageNet.

Compared with vanilla training, LRP only incurs a small accuracy loss of less than 0.5%. Besides, compared with the best accuracy obtained from the two schemes of vanilla training, ada-LRP increases the accuracy by 0.15%, and 0.31% on MNIST and CIFAR-10 respectively. Ada-LRP also reduces the accuracy loss suffered by LRP from 0.25% to less than 0.01% on CIFAR-100. These results validate that personalizing local learning rates can suppress the drift issue and thus improve the model accuracy, even if the learning rates are uniformly perturbed. Moreover, Table 2 also shows that ada-LRP increases the top-1 accuracy by 0.14% and reduces the top-5 accuracy by 0.06% compared with vanilla training (Scheme I) on ImageNet. Our experimental results of more types of non-IID data splits are reported in the Appendix.

5.3. Defense Results

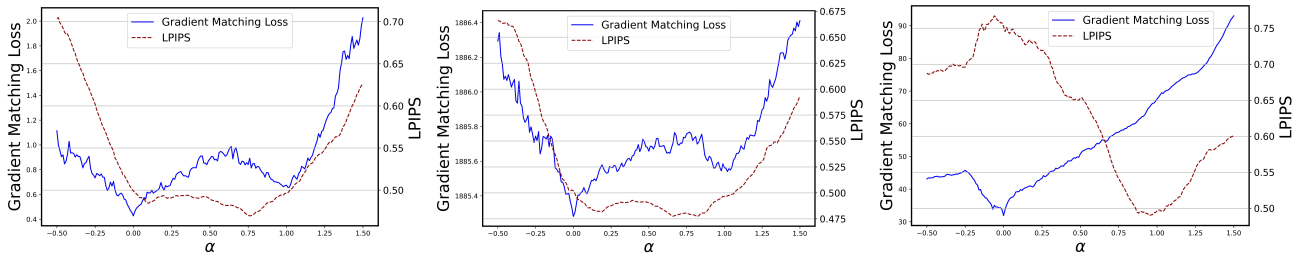
We choose an ideal case for the adversary where both the batch size and the number of local steps are set to 1. We evaluate our defenses in that case for a fair comparison with prior works [28, 35, 43]. But our defenses should provide much better privacy preservation in general cases (i.e., both B and E are larger than one). Since LRP and ada-LRP share the same key idea that uniformly perturbing the learning rates and hiding the learning rate expectations, we set $p_k N = \mathcal{K} p_k N = 2$, which is realistic for clients in a FL system, and thus unify the defense of the two methods. Results with other expectations can be found in the Appendix. We present visual results against DLG and iDLG with and without our defense on MNIST, CIFAR-100, and LFW in Figure 2. The results demonstrate our defense prevents data leakage under the two attack methods.

For defending GGL, we evaluate the similarity between the client's private images and their reconstructions via five types of evaluation metrics: Mean Square Error (MSE); Peak Signal-to-Noise Ratio (PSNR); Learned Perceptual Image Patch Similarity [41] measured by VGG network [34] (LPIPS-VGG) and ALEX network [22] (LPIPS-ALEX); MSE in Representation Space [28] (MSE-R). Due to the difference of random seeds, our results of the baselines are similar but not identical to the original paper [28].

From the visualization results in Figure 3, it can be seen that GGL produces high-quality images that can reveal attributes of the original data when the baseline defenses are applied. Moreover, the reconstructions of GGL become images that seem to be arbitrarily generated by GAN under our defense setting. For the quantitative comparison, Table 3 shows that our defense statistically outperforms the baseline defenses for all the evaluation metrics.

Experimental results on more privacy defense strategies and attack methods are shown in the Appendix.

To investigate the data leakage problem under the model inversion attack, we visualize the gradient matching loss and the LPIPS in the GAN latent space, inspired by [28]. Specifically, we plot the loss functions by interpolating be-



(a) No defense (b) Additive noise (c) Learning rate perturbation

Figure 4: Curves of gradient matching loss and LPIPS under no defense, additive noise, and LRP.

tween the latent vectors z_1 found by GGL, which generate latent vectors based on the gradients, and z_2 from GAN inversion [15], which generate latent vectors giving the ground truth image, as $z(\alpha) = (1-\alpha)z_1 + \alpha z_2$. We plot the gradient matching loss and LPIPS w.r.t. α in Figure 4 in cases of no defense, additive noise, and LRP are applied respectively. We only show the result with the defense of additive noise while we have similar observations with cases where other baseline defenses are applied. We also extend the visualization to a 2D surface by adding a second direction vector. Details can be found in the Appendix.

It can be seen from Figure 4 that: (1) In all three cases, the latent vector found by GGL reaches the lowest gradient matching loss on the curve. (2) When additive noise is applied, the gradient matching loss significantly increases. But the shape of the two curves in Figure 4b do not obviously change compared with the curves in Figure 4a. (3) In Figure 4a and 4b, the shapes of the two curves match well. The latent vector with the lowest gradient matching loss generates images with low LPIPS that can reveal private information of original data. (4) LRP reforms the two curves. The latent space vectors with low gradient matching loss do not result in images with low LPIPS, which explains why our defense enhances privacy preservation against the gradient inversion attack.

5.4. Convergence Results

To verify our analysis in section 4.2, we conduct experiments on MNIST in a non-IID setting as mentioned in section 5.1. As shown in Figure 5, LRP and ada-LRP converge well. It also can be seen that perturbing local learning rates has little effect on the convergence process, and personalizing clients' learning rates performs better than the original schemes, no matter whether the learning rates are perturbed or not, which cross-validate the results of Table 1. For fair comparisons, we choose FedAvg (Scheme I) as the vanilla training algorithm to eliminate the impact of different aggregation schemes. Details are shown in the Appendix.

6. Discussion

Our proposed defenses perturb the learning rates through sampling from uniform distributions. In theory, adversaries could nullify the defenses by brute-force guessing the learning rates. However, in realistic cases with a considerable number of local steps, adversaries would need to guess the learning rate values for each step. Since clients do not provide learning rate expectations, while theoretically possible, attempting to neutralize our defenses through learning rate guessing would significantly increase the computation overhead of the adversaries.

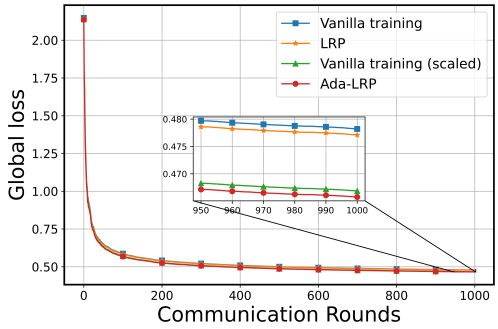


Figure 5: Convergence of LRP and ada-LRP. Vanilla training (scaled) denotes the case where the learning rates are personalized and not perturbed.

Beyond image recognition tasks, LRP can also enhance privacy protection against attacks on text.. Detailed experimental results can be found in Appendix.

7. Conclusion and Future Work

This work presents learning rate perturbation (LRP), which perturbs the local learning rates to enhance privacy preservation in FL. We also propose an adaptation (ada-LRP) for classification tasks to suppress the drift issue by personalizing clients' learning rate expectations. Our defenses offer stronger privacy protection compared with baselines, while the FL model's performance is maintained.

In this work, we uniformly perturb the learning rates. There are other methods for learning rate perturbation such as adding Gaussian and Laplacian noise, while adding noise following these distributions may generate extremely large learning rates, which is unacceptable for the training process. In the future, we will research on the impact of other random distributions.

Acknowledgments

This work was supported by Beijing University of Posts and Telecommunications-China Mobile Research Institute Joint Innovation Center and the Fundamental Research Funds for the Central Universities (2022RC14).

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016. 1, 2, 6

[2] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017. 1, 2, 5

[3] Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International Journal of Medical Informatics*, 112:59–67, 2018. 1

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018. 6

[5] Olivia Choudhury, Yoonyoung Park, Theodoros Salonidis, Aris Gkoulalas-Divanis, Issa Sylla, et al. Predicting adverse drug reactions on distributed health data using federated learning. In *AMIA Annual Symposium Proceedings*, volume 2019, page 313. American Medical Informatics Association, 2019. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 2

[7] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 6

[8] Whitfield Diffie and Martin E Hellman. New directions in cryptography. In *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, pages 365–390. 2022. 5

[9] Liam Fowl, Jonas Geiping, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models. *arXiv preprint arXiv:2110.13057*, 2021. 1, 2

[10] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020. 1, 2

[11] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017. 6

[12] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*, 2017. 2

[13] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008. 2

[14] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems*, 34:7232–7241, 2021. 1, 2, 3

[15] Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images into class-conditional generative networks. In *European Conference on Computer Vision*, pages 17–34. Springer, 2020. 8

[16] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021. 1, 2

[17] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 2, 3

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[19] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016. 1

[20] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. 1

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 7

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[24] Khaled B Letaief, Wei Chen, Yuanming Shi, Jun Zhang, and Ying-Jun Angela Zhang. The roadmap to 6G: AI empowered wireless networks. *IEEE Communications Magazine*, 57(8):84–90, 2019. 1

[25] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-IID data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978. IEEE, 2022. 6

[26] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. *arXiv preprint arXiv:1907.02189*, 2019. 3, 4, 6

[27] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021. 2

[28] Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10132–10142, 2022. 1, 2, 3, 6, 7

[29] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017. 1, 2, 6

[30] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989. 6

[31] Yi Liu, Xingliang Yuan, Zehui Xiong, Jiawen Kang, Xiaofei Wang, and Dusit Niyato. Federated learning for 6G communications: Challenges, methods, and future directions. *China Communications*, 17(9):105–118, 2020. 1

[32] Jiahao Lu, Xi Sheryl Zhang, Tianli Zhao, Xiangyu He, and Jian Cheng. APRIL: Finding the achilles' heel on privacy for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10051–10060, 2022. 1, 2

[33] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 3, 6

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[35] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9311–9319, 2021. 1, 2, 6, 7

[36] Paul Voigt and Axel Von dem Bussche. The EU general data protection regulation (GDPR). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017. 1

[37] Wensi Yang, Yuhang Zhang, Kejiang Ye, Li Li, and Cheng-Zhong Xu. FFD: A federated learning based method for credit card fraud detection. In *International Conference on Big Data*, pages 18–32. Springer, 2019. 1

[38] Zhaohui Yang, Mingzhe Chen, Kai-Kit Wong, H Vincent Poor, and Shuguang Cui. Federated learning for 6G: Applications, challenges, and opportunities. *Engineering*, 2021. 1

[39] Andrew C Yao. Protocols for secure computations. In *Proceedings of the Annual Symposium on Foundations of Computer Science*, pages 160–164. IEEE, 1982. 1

[40] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021. 2

[41] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 7

[42] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. iDLG: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020. 2, 6

[43] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2, 6, 7