Optimizing Privacy-Accuracy Trade-off in DP-FL via Significant Gradient Perturbation

Benteng Zhang[b], Yingchi Mao[a, b], Zijian Tu[b], Xiaoming He[b], Ping Ping[a, b], and Jie Wu[c]
[a] Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing, China
[b] School of Computer and Information, Hohai University, Nanjing, China
[c] Center for Networked Computing, Temple University, Philadelphia, USA
211307050017@hhu.edu.cn, yingchimao@hhu.edu.cn, 201307040020@hhu.edu.cn
isxmhe@gmail.com, pingpingnjust@163.com, jiewu@temple.edu

***Abstract*—In federated learning with differential privacy, an obvious phenomenon of local gradient sparsity emerges in some training rounds. When training with low privacy budgets, there is a risk of excessive noise being added to the uploaded gradients, leading to a significant decrease in the accuracy of the global model. To tackle the trade-off between privacy protection and model accuracy with low privacy budgets, we propose a differential privacy federated aggregation method based on gradient sparsity (DP-FedAGS), which not only prevents excessive noise addition by protecting only significant gradients, but also accelerates global model convergence by dynamically calculating the weight of the gradient. Experimental results indicate that DP-FedAGS achieves comparable privacy protection to DP-FedAvg and cpSGD, while outperforming DP-FedSNLC. Moreover, our approach respectively attains an approximate average test accuracy improvement of 2.45%, 4.79%, and 0.29% over the above three methods, rendering DP-FedAGS a promising approach for exploring a balance between privacy protection and model accuracy.**

***Index Terms*—Federated Learning, Differential Privacy, Gradient Sparsity, Aggregation Weights**

I. Introduction

Differential Privacy (DP) [1] is widely employed in Federated Learning (FL) [2]. By introducing random noise to the gradient before uploading, DP protects the privacy information of the client. The definition of DP incorporates the privacy budget ε (a non-negative real number) as a parameter used to quantify the level of privacy protection. A smaller ε indicates a higher degree of desired privacy protection. Satisfying the definition of ε-DP [3] ensures that the processing of individual information in the data set is done with a specified level of privacy protection, thereby mitigating the risk of privacy disclosure. DP is frequently combined with local gradient sparsity [4] to reinforce privacy protection. Nonetheless, in the context of DP-FL, the phenomenon of local gradient sparsity becomes particularly prominent in some training rounds. When training with low privacy budgets, gradient sparsity can give rise to an abundance of noise in the uploaded gradient, leading to gradient distortion and a decline in the accuracy of the global model [5]. Therefore, our objective is to explore the trade-off between privacy protection and model accuracy in DP-FL. Balancing privacy protection and model accuracy has long been a formidable challenge in DP-FL [6]. Some researchers endeavor to tackle this challenge in two ways:

(1) Sparsity and quantization. These methods entail uploading gradients from clients that possess limited parameter information, thereby diminishing the likelihood of detailed gradient information leakage and subsequently alleviating the risk of sensitive data exposure. Mao et al. [7] proposed a communication-efficient FL framework that dynamically adapts the quantization levels based on local gradient updates. SDGM [8] integrates sparsification techniques with Gaussian noise to provide privacy guarantees for the centralized SGD algorithm. Lyu et al. [9] proposed DP-SIGNSGD, an efficient, privacy-preserving, and Byzantine-robust compression algorithm rooted in the concept of gradient sparsification. Hu et al. [10] combined gradient perturbation with random sparsification and presented the Fed-SPA method, which bolsters privacy through sparsification. cpSGD [11] adds noise after gradient quantization, and it is commonly used as a baseline method in many works. Nonetheless, sparsity and quantization reduce the accuracy of the gradient, while high-dimensional vector quantization imposes higher costs, potentially leading to a decline in the accuracy of the trained global model.

(2) Protecting significant gradients. Significant gradients usually encompass vital updates of model parameters, and protecting these gradients can alleviate the influence of noise addition on model accuracy, guaranteeing that the model sustains a high level of accuracy while ensuring privacy. Recent research [12] indicates that the majority of gradient values updated by clients are close to zero. Consequently, in the context of low privacy budgets, clients should solely safeguard significant gradients (those far from zero) to mitigate privacy budget consumption. DP-ADMM [13] utilizes the alternating direction method of multipliers for iterative convergence but is restricted to convex functions. DP-FedSNLC [14] evaluates the significance of gradients by assessing changes in the loss function and applies noise perturbation to the significant gradients. However, these studies exhibit limitations concerning the clients, models, and datasets employed in FL to achieve a balance between privacy protection and model accuracy, which do not align with the desired trade-off approach. In conclusion, formulating a DP-FL method that prioritizes significant gradients to precisely capture the trade-off between privacy protection and model accuracy remains a substantial challenge in this domain.

Motivated by the above problems, we combine the ideas of gradient sparsity and protecting significant gradients to propose DP-FedAGS, which tackles the trade-off between privacy protection and model accuracy with low privacy budgets. In general, DP-FedAGS prevents excessive noise addition by only protecting significant gradients and accelerating global model convergence by calculating dynamic aggregation weights for the gradients. The main contributions of this paper are as follows:

- We prove that introducing Laplace noise into partial significant gradients successfully satisfies the definition of ε-DP.
- To mitigate excessive noise addition when training with low privacy budgets, we introduce a threshold calculation method to assess and protect significant gradients.
- To accelerate global model convergence, we propose a dynamic gradient aggregation method to dynamically calculate gradient weights and aggregate global gradients.
- Experiments on the MNIST, CIFAR-10, and CIFAR-100 datasets demonstrate that DP-FedAGS effectively improves the accuracy and availability of the global model while ensuring privacy protection.

The remainder of this paper is organized as follows. Section 2 describes the system model. Partial gradient Laplace noise based on ε-DP is proved in Section 3. The design details of DP-FedAGS are discussed in Section 4. The experiments and analysis are given in Section 5. At last, conclusions are drawn in Section 6.

II. System Model

As shown in Fig. 1, the system model comprises a central server and K clients. Each client has its local privacy dataset D_k and collectively train a global model with parameters W while ensuring its protection with DP. Client k iterates locally for E times to update its local model M_k and introduces noise N to the significant gradients. Subsequently, the client uploads the processed gradient $g_t(k)$, local model training loss $L_k(W_t)$, and local data size n_k to the server. The server computes the gradient aggregation weight $\gamma_t(k)$ and aggregates the global gradient g_t by considering $g_t(k)$ and $\gamma_t(k)$. After that, the server updates the global model parameters W_{t+1}. These steps are iterated until the global model converges and attains the desired performance.

After T training rounds, the noise added to the gradients will be scaled to $N(0, \sigma^2 C^2 I)$. With low privacy budgets, it will lead to higher noise level σ. Due to the sparsity of local gradients in some training rounds, different importance levels of gradients are treated differently using sparse vector techniques. In each training round, gradients greater than threshold λ_t will be perturbed, while the remaining gradients retain their original values. The perturbation method for the gradients and σ are given by

$$g_t(k) = \begin{cases} g'_t(k) + N(0, \sigma^2 C^2 I) & \text{if } g'_t(k) + \alpha \geq \lambda_t + \beta \\ g'_t(k) & \text{otherwise} \end{cases}, \tag{1}$$

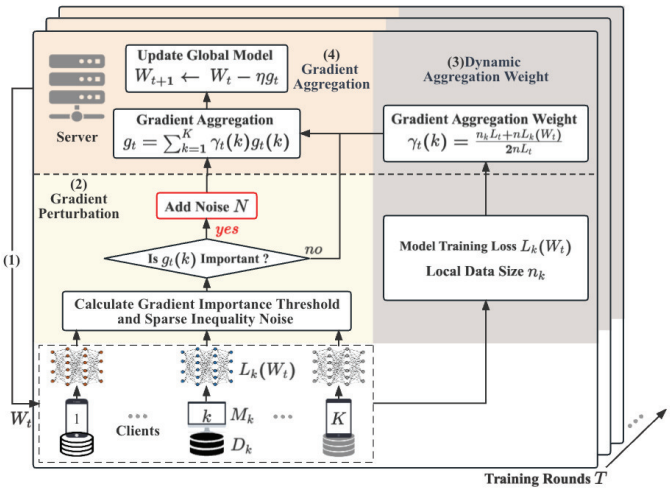

Fig. 1. The framework of DP-FedAGS.

$$\sigma = \frac{\Delta S}{\varepsilon} \sqrt{2 \ln(\frac{1.25}{\zeta})}, \tag{2}$$

where $g_t(k)$ is the gradient uploaded by the k-th client in the t-th training round, $g'_t(k)$ is the locally clipped gradient, ΔS is global sensitivity and $\zeta = e^{-\varepsilon}$ is noise level. At the same time, the weights for aggregating the gradients uploaded by each client are often fixed on the server side. However, when the local data of clients are equal and not independently identically distributed (Non-IID), perturbing partial gradients and aggregating the global model based on FedAvg [15] will result in slower convergence of the global model. Therefore, we modified the aggregation method of the global model to ensure that the server can quickly and accurately achieve the predefined global objective, which is given by

$$g_t = \sum_{k=1}^{K} \gamma_t(k) g_t(k), \tag{3}$$

where $\gamma_t(k) \in [0, 1]$, $\sum \gamma_t(k) = 1$ and g_t is the global aggregated gradient in the t-th training round. Since our method only perturbs partial significant gradients with noise, which changes the definition conditions of Laplace noise for DP. Therefore, it is necessary to rigorously prove whether DP-FedAGS satisfies ε-DP, i.e., to reevaluate its compliance with the following definition:

$$\Pr[M(\mathbf{D}) = O] \leq e^{\varepsilon} \Pr[M(\mathbf{D}') = O], \tag{4}$$

where $\Pr[\cdot]$ is the probability, $M(\mathbf{D}) = (x_1, ..., x_{wd}, ..., x_d)^T$, $M(\mathbf{D}') = (x_1 + \Delta x_1, ..., x_{wd} + \Delta x_{wd}, ..., x_d)^T$ and O is the output vector.

III. Partial Gradient Laplace Noise Based on ε-DP

In order to provide more rigorous and better privacy protection and facilitate the combined use of various DP mechanisms, we opt for Laplace noise [16] as the perturbation source, which satisfies the ε-DP definition. Contrasted with Gaussian noise [17], Laplace noise delivers more stringent

TABLE I
LIST OF MAIN SYMBOLIC PARAMETERS

Symbol	Symbol Meaning
K	Number of Clients
σ	Noise Standard Deviation
ε	Privacy Budget
d	Dimensions of Global Model
η	Learning Rate
ζ	Noise Level
ω	Gradient Selection Coefficient
δ	Relaxation Term of Noise
n_k	Local Data Size of Client k
λ_t	Threshold for Perturbing Gradients
g_t	Global Gradient in t-th Iteration
N	Noise
M	Global Model
T	Training Rounds
E	Local Iterations
B	Local Batch Size
C	Fixed Clipping Threshold
W	Global Model Parameters
D_k	Local Privacy Dataset
L_t	Total Model Training Loss of Clients
ΔS	Global Sensitivity
$\gamma_t[k]$	Aggregation Weight of Client k
α, β	Noise for Evaluating Query Results
D, D'	Sibling Datasets
$L_k(W_t)$	Loss Function for Client k

privacy safeguards at the expense of compromising information accuracy. However, DP-FedAGS changes the definition conditions of Laplace noise for DP. This section will start with the definition of ε-DP and discuss how our method satisfies Laplace-DP for partial gradients. Our goal is to prove that adding Laplace noise to partial significant gradients can satisfy the requirements of the ε-DP definition.

Definition 1. The probability density function of the Laplace distribution for the random variable x is defined as (5). The parameter μ is the location of the added noise, while the variance is given by $2b^2$.

$$Lap(x \mid \mu, b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}. \tag{5}$$

Definition 2. The general definition of DP is as follows: Given a pair of sibling datasets $\mathbf{D}$ and $\mathbf{D}'$, for a function F_{model}:$\mathbf{D} \to \mathbb{R}^d$ that represents the mapping relationship from dataset $\mathbf{D}$ to a d-dimensional space, it has a sensitivity ΔS.

Definition 3. In order to satisfy the ε-DP definition with Laplace-distributed noise $Laplace_d(\frac{\Delta S}{\varepsilon})$, for any domain function with input X, it is defined as (6). The scale parameter of the Laplace distribution is $\frac{\Delta S}{\varepsilon}$.

$$F_{model}(X) + Laplace_d(\frac{\Delta S}{\varepsilon}). \tag{6}$$

Assumption 1. Let the input be an arbitrary domain function of $\mathbf{D}$, which is given by

$$F_{model}(\mathbf{D}) = (x_1, x_2, ..., x_d)^T. \tag{7}$$

After adding Laplace noise, the resulting output function is

$$\begin{aligned} &F'_{model}(\mathbf{D}) = F_{model}(\mathbf{D}) + \\ &(Laplace_1(\frac{\Delta S}{\varepsilon}), Laplace_2(\frac{\Delta S}{\varepsilon}), ..., Laplace_d(\frac{\Delta S}{\varepsilon})), \end{aligned} \tag{8}$$

where $\Delta S = \max_{\mathbf{D},\mathbf{D}'} \|F_{model}(\mathbf{D}) - F_{model}(\mathbf{D}')\|_p$, p is typically set to 1, and its specific representation is given by

$$\Delta S = \max_{\mathbf{D},\mathbf{D}'} (\sum_{i=1}^{d} |\Delta x_i|). \tag{9}$$

Because the output function $F'_{model}(\mathbf{D})$ satisfies the definition of DP, then we have

$$\Pr\left[F'_{model}(\mathbf{D}) = O\right] \leq e^{\varepsilon} \Pr\left[F'_{model}(\mathbf{D}') = O\right]. \tag{10}$$

Now, we need to prove the validity of (10) in order to prove that adding Laplace noise to partial significant gradients satisfies the definition of ε-DP.

Assumption 2. Assuming that we aggregate the global gradient based on the gradient weights, then we have

$$\begin{aligned} F_{model}(\mathbf{D}') &= (x'_1, x'_2, ..., x'_d)^T \\ &= (x_1 + \Delta x_1, x_2 + \Delta x_2, ..., x_d + \Delta x_d)^T, \end{aligned} \tag{11}$$

according to (11) we can get

$$\Delta S = \max_{\mathbf{D},\mathbf{D}'} (\sum_{i=1}^{d} \left|x_i - x'_i\right|). \tag{12}$$

We define ω as the gradient selection coefficient, where $\omega \in [0, 1]$. As $\omega \to 1$, more gradients are selected. Thus, for any domain function with inputs $\mathbf{D}$ and $\mathbf{D}'$, we have

$$F_{model}(\mathbf{D}) = (x_1, x_2, \ldots, x_{\omega d}, \ldots, x_d)^T, \tag{13}$$

$$\begin{aligned} F_{model}(\mathbf{D}') &= (x'_1, \ldots, x'_{\omega d}, \ldots, x'_d)^T \\ &= (x_1 + \Delta x_1, \ldots, x_{\omega d} + \Delta x_{\omega d}, \ldots, x_d)^T, \end{aligned} \tag{14}$$

then, with input $\mathbf{D}$, $\mathbf{D}'$ and ΔS, we can get

$$\begin{aligned} \Delta S_N &= \max_{\mathbf{D},\mathbf{D}'} (\sum_{i=1}^{\omega d} \mid x_i - x'_i \mid) \\ &= \max_{\mathbf{D},\mathbf{D}'} (\sum_{i=1}^{\omega d} \mid \Delta x_i \mid) \leq \Delta S. \end{aligned} \tag{15}$$

Assumption 3. Without loss of generality, we assume that all x_i in the input $\mathbf{D}$ are equal to 0. In this case, we have $F_{model}(\mathbf{D}) = (0, 0, ..., 0)^T$, $F_{model}(\mathbf{D}') = (\Delta x_1, \Delta x_2, ..., \Delta x_{wd}, ..., 0)^T$. When $O = (y_1, y_2, ..., y_d)^T$, we have

$$\Pr[F'_{model}(\mathbf{D}) = O] = \prod_{i=1}^{\omega d} \frac{\varepsilon}{2\Delta \mathbf{S}_N} e^{-\frac{\varepsilon}{\Delta S_N}|\gamma_i|}, \tag{16}$$

$$\Pr[F'_{model}(\mathbf{D}') = O] = \prod_{i=1}^{\omega d} \frac{\varepsilon}{2\Delta S_N} e^{\frac{\varepsilon}{\Delta S_N}|\Delta x_i - y_i|}, \tag{17}$$

then we can get

$$\begin{aligned} \frac{\Pr[F'_{model}(\mathbf{D}) = O]}{\Pr[F'_{model}(\mathbf{D}') = O]} &= \frac{\prod_{i=1}^{\omega d} \frac{\varepsilon}{2\Delta S_N} e^{-\frac{\varepsilon}{\Delta S_N}|y_i|}}{\prod_{i=1}^{\omega d} \frac{\varepsilon}{2\Delta S_N} e^{-\frac{\varepsilon}{\Delta S_N}|\Delta x_i - y_i|}} \\ &= e^{\frac{\varepsilon}{\Delta S_N} \sum_{i=1}^{\omega d} (|\Delta x_i - y_i| - |y_i|)}. \end{aligned} \tag{18}$$

Now, we need to prove $\sum_{i=1}^{\omega d}(|\Delta x_i - y_i| - |y_i|) \leq \Delta S_N$ in order to prove that (10) holds. For each $|\Delta x_i - y_i| - |y_i|$, according to the absolute inequality, we have

$$\sum_{i=1}^{\omega d}(-|\Delta x_i|) \leq \sum_{i=1}^{\omega d}(|\Delta x_i - y_i| - |y_i|) \leq \sum_{i=1}^{\omega d}(|\Delta x_i|), \quad (19)$$

and because

$$\sum_{i=1}^{\omega d}(|\Delta x_i|) \leq \max_{\mathbf{D},\mathbf{D}'}(\sum_{i=1}^{\omega d}|\Delta x_i|) = \Delta S_N \leq \Delta S, \quad (20)$$

according to (16), (17) and (20), we can get

$$\sum_{i=1}^{\omega d}(|\Delta x_i - y_i| - |y_i|) \leq \Delta S_N \leq \Delta S. \quad (21)$$

We can get $\sum_{i=1}^{\omega d}(|\Delta x_i - y_i| - |y_i|) \leq \Delta S_N$ from (21), which allows us to prove the validity of (10), i.e. $\Pr[F'_{model}(\mathbf{D}) = O] \leq e^{\varepsilon}\Pr[F'_{model}(\mathbf{D}') = O]$ holds. Therefore, we have proven the following theorem.

Theorem 1. Adding Laplace noise to partial significant gradients satisfies the definition of ε-DP and ensures the privacy of gradients.

IV. Gradient Perturbation and Aggregation

A. Gradient Perturbation Mechanism

According to ***Theorem 1***, it can be concluded that adding Laplace noise to partial significant gradients ensures the privacy of the gradients. As shown in Fig. 2, for client k, after computing the local λ_t, noise is added to the query results that exceed λ_t in d queries. We combine DP with Laplace noise, referred to as (ε, δ)-DP. When $\delta = 0$, random algorithm $\mathcal{A}$ satisfies ε-DP definition. We make the following assumptions:

Assumption 4. In FL, we assume that random algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfy ε_1-DP and ε_2-DP respectively, for sequentially executed algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$, they satisfy $(\varepsilon_1 + \varepsilon_2)$-DP.

Then, with the random algorithm $\mathcal{A}$, $S \subseteq \mathrm{Range}(\mathcal{A})$ and input $\mathbf{D}$, $\mathbf{D}'$, its formal definition is given by

$$\Pr[\mathcal{A}(\mathbf{D}) \in S] \leq e^{\varepsilon}\Pr[\mathcal{A}(\mathbf{D}') \in S] + \delta. \quad (22)$$

Assumption 5. The precise query result of input x on dataset $\mathbf{D}$ is represented as $R(x, \mathbf{D})$, and $\mathbf{N}$ is noise that follows Laplace distribution. The query result Q with Laplace noise added to satisfy ε-DP is given by

$$Q = R(x, \mathbf{D}) + \mathbf{N}. \quad (23)$$

Then, let $Lap(\Delta S/\varepsilon)$ denote the Laplace noise $\mathbf{N}$ that satisfies the ε-DP definition, which is defined by

$$\Pr(\mathbf{N}) = \frac{\varepsilon}{2\Delta S}e^{-\frac{\varepsilon}{\Delta S}|\mathbf{N}|}. \quad (24)$$

However, the above reasoning only considers the privacy budget consumed by a single query. According to the composition theorem of DP mechanisms satisfying the Laplace distribution in FL, if multiple queries are executed simultaneously, the privacy budget consumed will increase linearly.

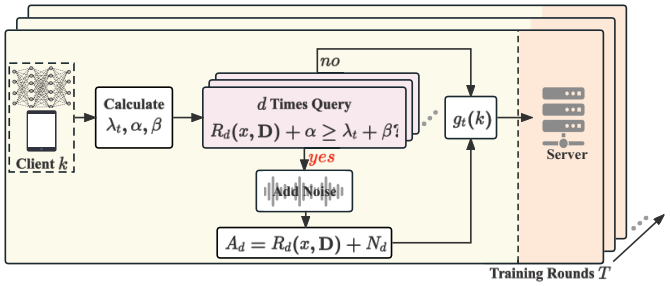

Fig. 2. Gradient perturbation of client k.

Without loss of generality, let d denote the dimension of the FL model, and the update of d parameters by a single client is equivalent to answering d queries simultaneously. Let x denote the input parameters of the d queries. The accurate query result of input x on dataset $\mathbf{D}$ is denoted as $R(x, \mathbf{D}) \in \mathbb{R}^d$, and the query result with Laplace noise satisfying ε-DP is denoted as $Q(x, \varepsilon)$, which is given by

$$Q(x, \varepsilon) = R(x, \mathbf{D}) + \mathbf{N}(\varepsilon). \quad (25)$$

To reduce the privacy budget consumption of simultaneously executing multiple queries, we introduce the idea of sparse vectors [18]. Laplace noise is only added when the queried content is deemed significant; otherwise, no operation is performed. Specifically, in a certain training round, if d queries are requested, Laplace noise is added only when $R_d(x, \mathbf{D}) + \alpha \geq \lambda + \beta$, then we have

$$A_d = R_d(x, \mathbf{D}) + N_d, \quad (26)$$

where A_d is the query result after applying Laplace noise perturbation for query d. λ is the threshold for determining the importance of the queried content, α and β are additional noise that evaluate the importance of the query result, following Laplace noise distributions $Lap(q\Delta S/\varepsilon_1)$ and $Lap(q\Delta S/\varepsilon_2)$, respectively. N_d is the noise used to perturb the query result, following Laplace noise distribution $Lap(q\Delta S/\varepsilon_3)$. However, the premise of using the above gradient perturbation method is that the total privacy budget satisfies $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$. Therefore, we need to prove $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$.

Assumption 6. When $\forall_i R_i(\mathbf{D}) \geq R_i(\mathbf{D}')$, (27) and (28) exist. Let κ denote the parameter input for the function $f_i(\mathbf{D}, \kappa)$.

$$f_i(\mathbf{D}, \kappa) = \Pr[R_i(\mathbf{D}) + \alpha < \lambda + \kappa], \quad (27)$$

$$g_i(\mathbf{D}, \kappa) = \Pr[R_i(\mathbf{D}) + \alpha \geq \lambda + \kappa], \quad (28)$$

then we have

$$\begin{aligned} f_i(\mathbf{D}, \kappa) &= \Pr[R_i(\mathbf{D}) + \alpha < \lambda + \kappa] \\ &\leq \Pr[R_i(\mathbf{D}') + \alpha < \lambda + \kappa] \\ &= f_i(\mathbf{D}', \kappa), \end{aligned} \quad (29)$$

$$\begin{aligned}\mathbf{g}_i(\mathbf{D},\kappa) &= \Pr[R_i(\mathbf{D})+\alpha \geq \lambda+\kappa] \\ &\leq \Pr[R_i(\mathbf{D}')+\alpha+\chi \geq \lambda+\kappa] \\ &\leq e^{\varepsilon_1/q}\Pr[R_i(\mathbf{D}')+\alpha \geq \lambda+\kappa] \\ &= e^{\varepsilon_1/q} g_i(\mathbf{D}',\kappa),\end{aligned} \tag{30}$$

then we can get

$$\begin{aligned}&\Pr[M(\mathbf{D})] \\ &\leq \int_{-\infty}^{+\infty} \Pr[\kappa=\beta]\prod_{j\in i} f_j(\mathbf{D}',\kappa)\prod_{j\notin i} e^{\varepsilon_1/q} g_j(\mathbf{D}',\kappa)d\kappa \\ &\leq (e^{\varepsilon_1/q})^q \Pr[M(\mathbf{D}')] \leq e^{\varepsilon_1+\varepsilon_2}\Pr[M(\mathbf{D}')].\end{aligned} \tag{31}$$

Following the above steps, $\varepsilon = \varepsilon_1+\varepsilon_2+\varepsilon_3$ holds.

Assumption 7. When $\forall_i R_i(\mathbf{D}) \leq R_i(\mathbf{D}')$, $\forall_i R_i(\mathbf{D}) \geq R_i(\mathbf{D}')-\chi$, according to the above steps, we can infer the following:

$$\begin{aligned}f_i(\mathbf{D},\kappa-\chi) &= \Pr[R_i(\mathbf{D})+\alpha < \lambda+\kappa-\chi] \\ &\leq \Pr[R_i(\mathbf{D}')-\chi+\alpha < \lambda+\kappa-\chi] \\ &= f_i(\mathbf{D}',\kappa),\end{aligned} \tag{32}$$

$$\begin{aligned}g_i(\mathbf{D},\kappa-\chi) &= \Pr[R_i(\mathbf{D})+\alpha \geq \lambda+\kappa-\chi] \\ &\leq \Pr[R_i(\mathbf{D}')+\alpha \geq \lambda+\kappa-\chi] \\ &\leq e^{\varepsilon_1/q}\Pr[R_i(\mathbf{D}')+\alpha \geq \lambda+\kappa] \\ &= e^{\varepsilon_1/q} g_i(\mathbf{D}',\kappa),\end{aligned} \tag{33}$$

where χ is the change in the variable. As the independent variable changes from κ to $\kappa-\chi$, according to the ε-DP definition, we can get

$$\begin{aligned}&\Pr[M(\mathbf{D})] = \int_{-\infty}^{+\infty}\Pr[\kappa=\beta+\chi]\prod_{j\in i} f_j(\mathbf{D}',\kappa-\chi) \\ &\prod_{j\notin i} e^{\varepsilon_i/q} g_j(\mathbf{D}',\kappa-\chi)d\kappa \\ &\leq \int_{-\infty}^{+\infty} e^{\varepsilon_2}\Pr[\kappa=\beta]\prod_{j\in i} f_j(\mathbf{D}',\kappa)\prod_{j\notin i} e^{\varepsilon_i/q} g_j(\mathbf{D}',\kappa)d\kappa \\ &\leq (e^{\varepsilon_1/q})^q e^{\varepsilon_2}\Pr[M(\mathbf{D}')] = e^{\varepsilon_1+\varepsilon_2}\Pr[M(\mathbf{D}')].\end{aligned} \tag{34}$$

Following the above steps, $\varepsilon = \varepsilon_1+\varepsilon_2+\varepsilon_3$ still holds. Based on ***Assumption 6*** and ***Assumption 7***, i.e., when $\forall_i R_i(\mathbf{D}) \geq R_i(\mathbf{D}')$ or $\forall_i R_i(\mathbf{D}) \leq R_i(\mathbf{D}')$, we can always get $\varepsilon = \varepsilon_1+\varepsilon_2+\varepsilon_3$, therefore $\varepsilon = \varepsilon_1+\varepsilon_2+\varepsilon_3$ holds. In this way, we have proven the following theorem.

Theorem 2. The total privacy budget satisfies $\varepsilon = \varepsilon_1+\varepsilon_2+\varepsilon_3$.

Because ε_3 can affect the perturbed gradient values returned to the server, so $\varepsilon_3 \gg \varepsilon_1+\varepsilon_2$. If ε_3 is too small, it will significantly reduce the model accuracy in FL. Conversely, even if $\varepsilon_1+\varepsilon_2$ is very small, perturbation will only occur when selecting valid gradients. To make the perturbation more accurate, we aim to minimize the variances of α and β. When $\varepsilon_1+\varepsilon_2$ is a fixed value, the privacy budget ratio $\varepsilon_1 : \varepsilon_2 = \sqrt[3]{q^2} : 1$. Meanwhile, the threshold λ is used to determine the importance of the queried content. We incorporate the idea of the Top-k [19] method into the selection of λ, setting different thresholds for different training rounds. In the early training rounds, when the parameters change dramatically and there is more gradient information, a larger threshold λ is set to ensure fast convergence of the model. In the later training rounds, when the parameters tend to stabilize and there is less gradient information, a smaller threshold λ is set to reduce the consumption of the privacy budget. After T training rounds, the total number of model parameters is $|W|$. The calculation of the threshold λ_t in the t-th training round is given by

$$\lambda_t = \min(sort(g')[\left\lceil\frac{t|W|}{T}\right\rceil], sort(g')[\left\lceil\frac{9|W|}{10}\right\rceil]), \tag{35}$$

where $sort(\cdot)$ represents the sorting result in ascending order, and g' represents the locally clipped gradient.

B. Gradient Aggregation Mechanism

In FL, the most commonly referenced algorithm is FedAvg [15], in which the server aggregates the weights of the gradients uploaded by clients. These weights are typically fixed and determined based on the size of local training data. After T training rounds, the global model objective must satisfy the following:

$$\min\sum_{k=1}^{K}\frac{n_k}{n}L_k(W), \tag{36}$$

where $L_k(W)$ denotes the loss function used to train the local model of the client k, $n = \sum_{k=1}^{K} n_k$ is the total data size over all participating clients, and n_k is the local data size of client k. The impact of local loss on the global objective depends entirely on the size of the local data. We hope that the gradient aggregation weights on the server side can reflect the aggregated global model through the global loss function. However, the accuracy of the gradients decreases after noise is applied, especially when partial gradients are perturbed. The information contained in the gradients uploaded by each client may be completely inconsistent with the previous values. Based on the effectiveness of local training, we dynamically adjust the gradient aggregation weight γ in each training round. Assuming that the local model training loss function for the client k in the t-th training round is $L_k(W_t)$, the total model training loss for the client k is $L_t = \sum_{k=1}^{K} L_k(W_t)$. If the proportion of $L_k(W_t)$ to L_t is relatively large, it indicates that the gradients uploaded by that client do not reflect the current trend of the model parameters well. Considering the influence of the local loss contained in the global objective function, which depends entirely on the size of the local data, the gradient aggregation weight $\gamma_t(k)$ for client k in the t-th training round is given by

$$\gamma_t(k) = \frac{n_k L_t + nL_k(W_t)}{2nL_t}, \tag{37}$$

where $\sum_{k=1}^{K}\frac{n_k}{n} = 1 \bigcap \sum_{k=1}^{K}\frac{L_k(W_t)}{L_t} = 1$ always holds, the gradient aggregation weight $|\gamma_t| = \sum_{k=1}^{K}\gamma_t(k) = 1$ for each client in the t-th training round is always true. We adopt

SGD [20] algorithm to calculate gradient and the detailed training steps are shown in ***Algorithm 1***. The server performs T training rounds, resulting in an overall complexity of the DP-FedAGS algorithm of $\mathcal{O}(ET)$. However, considering that local iterations $E \ll T$, the overall complexity of the DP-FedAGS can be simplified to $\mathcal{O}(n)$.

Algorithm 1: DP-FedAGS

Input: K, D_k, M_k, M, B, E, σ, ΔS, $\mathcal{E}_1$, $\mathcal{E}_2$, C, η, T
Output: Global model M
Initialize M, M_k
for *each training round* $t \in T$ **do**
  for *each client* k *in parallel* **do**
    $W_k \leftarrow M_k$;
    $g_k, L_k, n_k \leftarrow clientTrain(W_k, D_k, T, t)$;
  end
  $L \leftarrow \sum L_k, n \leftarrow \sum n_k$
  $\gamma_k \leftarrow \frac{n_k L + n L_k}{2nL}$
  $g \leftarrow \sum \gamma_k g_k$
  $W_{t+1} \leftarrow W_t - \eta g$
end
function $clientTrain(W_k, D_k, T, t)$
begin
  $n \leftarrow D$
  for *each local epoch* $i \in E$ **do**
    $g_i \leftarrow \nabla L(W)$;
    $W \leftarrow W - \eta g_i$;
  end
  $g \leftarrow \sum g_i, L = \sum \Delta M(W)_i$
  $g' \leftarrow \frac{g}{\max(1, \frac{\|g\|_2}{C})}$
  $\alpha \leftarrow Lap(\frac{\Delta S}{\varepsilon_1}), \beta \leftarrow Lap(\frac{\Delta S}{\varepsilon_2})$
  $\lambda_t = \min(sort(g')[\lceil \frac{t|W|}{T} \rceil], sort(g')[\lceil \frac{9|W|}{10} \rceil])$
  if $g' + \alpha \geq \lambda + \beta$ **then**
    $g \leftarrow g' + N(0, \sigma^2 C^2 I)$
  else
    $g \leftarrow g'$
  end
  return g, L, n
end

V. Performance Evaluation

A. Experiment Settings

1) FL datasets and parameter settings. Our experiment consists of 100 clients participating in FL. We train deep learning models using the MNIST, CIFAR-10, and CIFAR-100 datasets. We set the batch size as 128, Dirichlet distribution parameter as 1, relaxation parameter δ as 0.001 and privacy budgets are set to $\{0.1, 0.2, 0.5\}$. The fixed clipping threshold C is set as 1, and 10 devices participate in training each round of communication. In the training rounds T of MNIST and CIFAR-10, CIFAR-100 are 200 and 500, respectively, and the learning rate η is initialized to 0.1 and 0.05, respectively.

2) Baselines. a) DP-FedAvg [21] is a commonly used baseline method, which introduces the FedAvg algorithm into DP. *b)* cpSGD [11] improves the accuracy of training models by combining gradient quantization and DP. *c)* DP-FedSNLC [14] adds noise to significant gradients to improve the accuracy of training models by evaluating the changes in local loss function.

3) Evaluation metrics. a) Privacy protection: We use classic membership inference attack methods (MIA [22], ML-Leaks [23], and White-box [24]) during the model training process. Note that lower accuracy of inference attacks in the experimental results indicates better privacy protection. *b)* Global model availability: Higher average accuracy in the experimental results indicates higher model training accuracy and better availability. Note that with low privacy budgets, we should pay particular attention to changes in the average accuracy of model training.

B. Privacy Protection

The experimental results for attack accuracy are presented in Table II. cpSGD achieves the lowest attack accuracy across different privacy budgets and inference attacks, indicating the strongest privacy protection effect. DP-FedAvg exhibits lower attack accuracy compared to DP-FedSNLC and DP-FedAGS, providing privacy protection second only to cpSGD and superior to DP-FedAGS. DP-FedSNLC demonstrates strong privacy protection capabilities during the initial stages of training; however, it experiences slower model updates in later stages, resulting in less effective defense against inference attacks compared to DP-FedAGS, particularly under ML-Leaks and White-box attacks. DP-FedAGS exhibits slightly higher attack accuracy compared to DP-FedAvg and cpSGD, but significantly lower than DP-FedSNLC. As DP-FedAGS only applies noise perturbation to a subset of significant gradients and adapts the weight of such gradients based on training rounds, its noise perturbation is lower compared to DP-FedAvg and cpSGD. Consequently, the defense effect of DP-FedAGS against inference attacks is slightly inferior to that of DP-FedAvg and cpSGD. Additionally, on the basis of Table II, we provide a summary of the average attack accuracy for the three attacks across varying privacy budgets and model training methods in Table III. Due to the low complexity of MNIST, there is little difference in the average attack accuracy among the four methods. However, as the complexity of the training dataset increases, the attack accuracy of the four methods significantly improves. With different datasets and privacy budgets, the average attack accuracy of DP-FedAGS is slightly higher than that of DP-FedAvg and cpSGD but much lower than that of DP-FedSNLC. Especially with low privacy budgets, when training models using complex datasets, DP-FedAGS outperforms DP-FedSNLC in privacy protection, and its privacy protection effectiveness is similar to that of DP-FedAvg and cpSGD.

C. Global Model Availability

We obtained experimental results in Fig. 3 from Table IV, which show that during MNIST training, when the privacy

TABLE II
EXPERIMENTAL RESULTS OF ATTACK ACCURACY OF DIFFERENT ATTACK MODELS WITH DIFFERENT TRAINING ALGORITHMS

Privacy Budget	Method (DP-)	MNIST			CIFAR-10			CIFAR-100		
		Basic MIA	ML-Leaks	White-box	Basic MIA	ML-Leaks	White-box	Basic MIA	ML-Leaks	White-box
0.1	FedAvg	50.01%	50.02%	50.02%	50.86%	51.42%	53.61%	51.45%	53.42%	55.01%
	cpSGD	50.01%	50.01%	50.02%	50.73%	51.31%	53.57%	51.24%	53.31%	54.96%
	FedSNLC	50.09%	50.21%	50.24%	51.38%	58.84%	62.59%	54.77%	60.26%	65.86%
	FedAGS	50.04%	50.09%	50.11%	50.89%	51.80%	54.38%	51.92%	54.18%	55.84%
0.2	FedAvg	50.04%	50.18%	50.35%	53.07%	55.12%	60.47%	55.63%	58.03%	61.76%
	cpSGD	50.03%	50.16%	50.23%	52.99%	55.06%	60.42%	55.49%	57.65%	61.64%
	FedSNLC	50.15%	50.27%	50.55%	55.56%	60.69%	67.19%	58.46%	65.88%	69.49%
	FedAGS	50.09%	50.21%	50.36%	53.58%	55.33%	60.70%	55.84%	58.91%	62.07%
0.5	FedAvg	50.13%	50.39%	50.83%	55.24%	59.32%	65.23%	60.29%	63.22%	68.10%
	cpSGD	50.10%	50.36%	50.81%	55.08%	59.25%	65.17%	60.02%	63.07%	67.70%
	FedSNLC	50.26%	50.52%	50.91%	58.91%	62.84%	70.36%	64.41%	70.15%	73.63%
	FedAGS	50.17%	50.44%	50.86%	55.45%	59.47%	65.60%	60.74%	64.06%	68.34%

TABLE III
EXPERIMENTAL RESULTS OF AVERAGE ATTACK ACCURACY OF DIFFERENT ATTACK MODELS WITH DIFFERENT TRAINING ALGORITHMS

Privacy Budget	Method (DP-)	Dataset		
		MNIST	CIFAR-10	CIFAR-100
		Average Accuracy	Average Accuracy	Average Accuracy
0.1	FedAvg	50.02%	51.96%	53.29%
	cpSGD	50.01%	51.87%	53.17%
	FedSNLC	50.18%	57.60%	60.30%
	FedAGS	50.08%	52.36%	53.98%
0.2	FedAvg	50.19%	56.22%	58.47%
	cpSGD	50.14%	56.16%	58.26%
	FedSNLC	50.32%	61.15%	64.61%
	FedAGS	50.22%	56.54%	58.94%
0.5	FedAvg	50.45%	59.93%	63.87%
	cpSGD	50.42%	59.83%	63.60%
	FedSNLC	50.56%	64.04%	69.40%
	FedAGS	50.49%	60.17%	64.38%

TABLE IV
EXPERIMENTAL RESULTS OF GLOBAL TEST ACCURACY WITH DIFFERENT PRIVACY BUDGETS FOR DIFFERENT TRAINING ALGORITHMS

Privacy Budget	Method (DP-)	Dataset		
		MNIST	CIFAR-10	CIFAR-100
		Average Accuracy	Average Accuracy	Average Accuracy
0.1	FedAvg	88.34%	34.49%	10.36%
	cpSGD	85.53%	31.92%	8.92%
	FedSNLC	90.74%	39.82%	13.94%
	FedAGS	91.16%	40.96%	14.34%
0.2	FedAvg	91.68%	42.88%	16.17%
	cpSGD	89.15%	39.51%	14.89%
	FedSNLC	92.92%	46.53%	18.91%
	FedAGS	93.24%	46.86%	19.44%
0.5	FedAvg	93.87%	48.36%	20.21%
	cpSGD	90.63%	45.94%	18.31%
	FedSNLC	94.28%	51.94%	22.15%
	FedAGS	94.75%	51.37%	22.63%

budget is set to 0.1, DP-FedAGS achieves a global test accuracy improvement of approximately 2.82% compared to DP-FedAvg. When the privacy budget is set to 0.5, DP-FedAGS achieves a global test accuracy improvement of approximately 0.88% compared to DP-FedAvg. Similarly, during CIFAR-10 and CIFAR-100 training, when the privacy budget is set to 0.1, DP-FedAGS exhibits a larger improvement in global test accuracy compared to DP-FedAvg, and the improvement is higher than when the privacy budget is set to 0.5. Therefore, DP-FedAGS can enhance the availability of global training models with low privacy budgets. The MNIST dataset is relative simple, which leads to minimal differences in accuracy among the four model training methods. However, on the moderately complex CIFAR-10 dataset, DP-FedAGS exhibits a substantial enhancement in global test accuracy, outperforming both DP-FedAvg and cpSGD. For complex datasets, DP-FedAGS selectively applies noise perturbation to important gradients during each training round and dynamically calculates the gradient aggregation weights. Consequently, DP-FedAGS can more precisely regulate the gradient perturbation in each round, particularly in complex datasets.

D. Simulation Summary

In summary, DP-FedAGS outperforms DP-FedSNLC in terms of privacy protection while remaining comparable to DP-FedAvg and cpSGD. For the relatively low-complexity dataset MNIST, there is minimal disparity in the global test accuracy between DP-FedAGS and the three baseline methods. However, when training with CIFAR-10 and CIFAR-100, DP-FedAGS exhibits superior performance in global test accuracy with low privacy budgets when compared to the three baseline methods. Therefore, when training with low privacy budgets, DP-FedAGS effectively addresses the trade-off between privacy protection and model accuracy.

VI. CONCLUSIONS

In this paper, we tackle the trade-off between privacy protection and model accuracy with low privacy budgets by proposing a novel approach DP-FedAGS, which protects only the significant gradients and prevents excessive noise addition, while accelerating the convergence of the global model. Experiments on MNIST, CIFAR-10, and CIFAR-100 manifest that our approach can more effectively perturb the significant gradients during each training round and dynamically calculate gradient aggregation weights based on the clients' local training. DP-FedAGS considerably enhances the accuracy and availability of model training while ensuring privacy protection, thereby effectively tackling the trade-off between privacy protection and model accuracy.

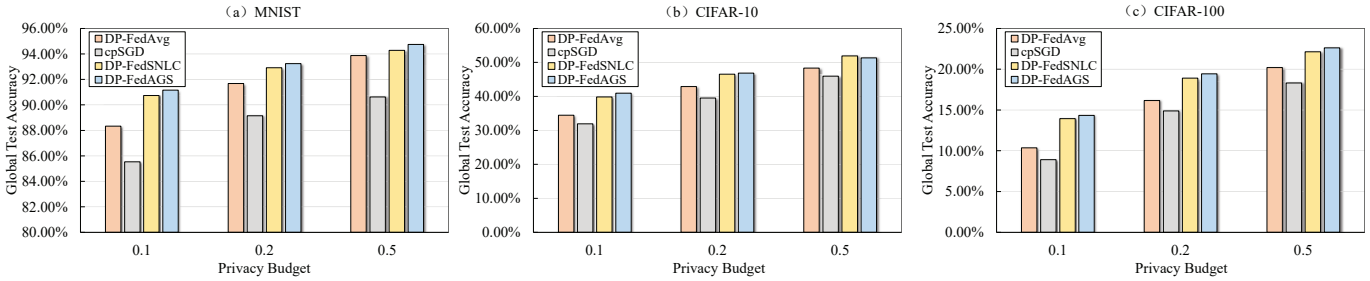

Fig. 3. The trend of changes in global test accuracy with different training methods and privacy budgets. As shown in (a), during MNIST training, DP-FedAGS achieves higher average global test accuracy than DP-FedAvg, cpSGD, and DP-FedSNLC, with improvements of approximately 1.75%, 4.61%, and 0.40% respectively. In (b), during CIFAR-10 training, DP-FedAGS outperforms DP-FedAvg, cpSGD, and DP-FedSNLC, with improvements of approximately 4.49%, 7.27%, and 0.30% respectively. In (c), during CIFAR-100 training, DP-FedAGS achieves higher global test accuracy than DP-FedAvg, cpSGD, and DP-FedSNLC, with improvements of approximately 3.22%, 5.55%, and 0.39% respectively.

ACKNOWLEDGMENT

This work is supported by The Key Research and Development Program of China (No. 2022YFC3005401), Key Research and Development Program of China, Yunnan Province (No. 202203AA080009) and Transformation Program of Scientific and Technological Achievements of Jiangsu Provence (No. BA2021002).

REFERENCES

[1] C. Dwork, "Differential Privacy," in Automata, Languages and Programming, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.

[2] A. B and S. S, "A survey on genomic data by privacy-preserving techniques perspective," Computational Biology and Chemistry, vol. 93, p. 107538, 2021, doi: https://doi.org/10.1016/j.compbiolchem.2021.107538.

[3] Y. Zhao and J. Chen, "A Survey on Differential Privacy for Unstructured Data Content," ACM Comput. Surv., vol. 54, no. 10s, Sep. 2022, doi: 10.1145/3490237.

[4] X. Qiu, J. Fernandez-Marques, P. P. Gusmao, Y. Gao, T. Parcollet, and N. D. Lane, "ZeroFL: Efficient On-Device Training for Federated Learning with Local Sparsity," 2022, doi: 10.48550/ARXIV.2208.02507.

[5] O. Shahid, S. Pouriyeh, R. M. Parizi, Q. Z. Sheng, G. Srivastava, and L. Zhao, "Communication Efficiency in Federated Learning: Achievements and Challenges," 2021, doi: 10.48550/ARXIV.2107.10996.

[6] A. E. Ouadrhiri and A. Abdelhadi, "Differential Privacy for Deep and Federated Learning: A Survey," in IEEE Access, vol. 10, pp. 22359-22380, 2022, doi: 10.1109/ACCESS.2022.3151670.

[7] Y. Mao et al., "Communication-Efficient Federated Learning with Adaptive Quantization," ACM Trans. Intell. Syst. Technol., vol. 13, no. 4, pp. 1–26, Aug. 2022, doi: 10.1145/3510587.

[8] X. Zhang, M. Fang, J. Liu, and Z. Zhu, "Private and communication-efficient edge learning: a sparse differential gaussian-masking distributed SGD approach," in Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, Virtual Event USA: ACM, Oct. 2020, pp. 261–270. doi: 10.1145/3397166.3409123.

[9] L. Lyu, "DP-SIGNSGD: When Efficiency Meets Privacy and Robustness," 2021, doi: 10.48550/ARXIV.2105.04808.

[10] R. Hu, Y. Gong, and Y. Guo, "Federated Learning with Sparsification-Amplified Privacy and Adaptive Optimization," 2020, doi: 10.48550/ARXIV.2008.01558.

[11] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpSGD: Communication-efficient and differentially-private distributed SGD," in Advances in Neural Information Processing Systems, Curran Associates, Inc., 2018. Accessed: Jul. 05, 2023. [Online].

[12] L. Cui, X. Su, Y. Zhou and Y. Pan, "Slashing Communication Traffic in Federated Learning by Transmitting Clustered Model Updates," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 8, pp. 2572-2589, Aug. 2021, doi: 10.1109/JSAC.2021.3087262.

[13] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin and Y. Gong, "DP-ADMM: ADMM-Based Distributed Learning With Differential Privacy," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1002-1012, 2020, doi: 10.1109/TIFS.2019.2931068.

[14] J. Ding, J. Wang, G. Liang, J. Bi, and M. Pan, "Towards Plausible Differentially Private ADMM Based Distributed Machine Learning," in Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event Ireland: ACM, Oct. 2020, pp. 285–294. doi: 10.1145/3340531.3411860.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR, Apr. 2017, pp. 1273–1282. Accessed: Jul. 06, 2023. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[16] J. Neri, P. Depalle and R. Badeau, "Approximate Inference and Learning of State Space Models With Laplace Noise," in IEEE Transactions on Signal Processing, vol. 69, pp. 3176-3189, 2021, doi: 10.1109/TSP.2021.3075146.

[17] Z. Chuanxin, S. Yi, and W. Degang, "Federated Learning with Gaussian Differential Privacy," in Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence, Shanghai China: ACM, Oct. 2020, pp. 296–301. doi: 10.1145/3438872.3439097.

[18] B. Shim, "Sparse Vector Coding for Ultra-reliable and Low-latency Communications," in Ultra-Reliable and Low-Latency Communications (URLLC) Theory and Practice, T. Duong, S. Khosravirad, C. She, P. Popovski, M. Bennis, and T. Quek, Eds., 1st ed.Wiley, 2023, pp. 169–213. doi: 10.1002/9781119818366.ch6.

[19] Q. Yang, X. Du, A. Liu, N. Wang, W. Wang, and X. Wu, "AdaSTopk: Adaptive federated shuffle model based on differential privacy," Information Sciences, vol. 642, p. 119186, 2023, doi: https://doi.org/10.1016/j.ins.2023.119186.

[20] J. Wang and G. Joshi, "Cooperative SGD: a unified framework for the design and analysis of local-update SGD algorithms," J. Mach. Learn. Res., vol. 22, no. 1, p. 213:9709-213:9758, Jan. 2021.

[21] M. Abadi et al., "Deep Learning with Differential Privacy," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna Austria: ACM, Oct. 2016, pp. 308–318. doi: 10.1145/2976749.2978318.

[22] Y. Gu, Y. Bai, and S. Xu, "CS-MIA: Membership inference attack based on prediction confidence series in federated learning," Journal of Information Security and Applications, vol. 67, p. 103201, Jun. 2022, doi: 10.1016/j.jisa.2022.103201.

[23] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models," 2018, doi: 10.48550/ARXIV.1806.01246.

[24] L. Song, R. Shokri, and P. Mittal, "Privacy Risks of Securing Machine Learning Models against Adversarial Examples," in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London United Kingdom: ACM, Nov. 2019, pp. 241–257. doi: 10.1145/3319535.3354211.