





Balancing Privacy and Accuracy Using Significant Gradient Protection in Federated Learning

Benteng Zhang , *Student Member, IEEE*, Yingchi Mao , *Member, IEEE*, Xiaoming He , *Member, IEEE*, Huawei Huang , *Senior Member, IEEE*, and Jie Wu , *Fellow, IEEE*

Abstract—Previous state-of-the-art studies have demonstrated that adversaries can access sensitive user data by membership inference attacks (MIAs) in Federated Learning (FL). Introducing differential privacy (DP) into the FL framework is an effective way to enhance the privacy of FL. Nevertheless, in differentially private federated learning (DP-FL), local gradients become excessively sparse in certain training rounds. Especially when training with low privacy budgets, there is a risk of introducing excessive noise into clients' gradients. This issue can lead to a significant degradation in the accuracy of the global model. Thus, how to balance the user's privacy and global model accuracy becomes a challenge in DP-FL. To this end, we propose an approach, known as differential privacy federated aggregation, based on significant gradient protection (DP-FedASGP). DP-FedASGP can mitigate excessive noises by protecting significant gradients and accelerate the convergence of the global model by calculating dynamic aggregation weights for gradients. Experimental results show that DP-FedASGP achieves comparable privacy protection effects to DP-FedAvg and cpSGD (communication-private SGD based on gradient quantization) but outperforms DP-FedSNLC (sparse noise based on clipping losses and privacy budget costs) and FedSMP (sparsified model perturbation). Furthermore, the average global test accuracy of DP-FedASGP across four datasets and three models is about 2.62%, 4.71%, 0.45%, and 0.19% higher than the above methods, respectively. These improvements indicate that DP-FedASGP is a promising approach for balancing the privacy and accuracy of DP-FL.

Index Terms—Federated learning, differential privacy, significant gradient protection.

I. INTRODUCTION

FEDERATED Learning (FL) [1] has gained substantial attention in the field of distributed machine learning frameworks. FL can protect users' privacy by enabling multiple parties to train machine learning models without sharing raw data. However, as illustrated in *Challenge 1* in Fig. 1, Hu et al. [2] demonstrated a persistent risk of privacy breaches in training data, due to the fact that external adversaries can infer sensitive user data characteristics by Membership Inference Attacks (MIAs). Differentially Private Federated Learning (DP-FL) introduces controlled random noise into the gradient before uploading to address this issue. Moreover, as depicted in *Challenge 2* in Fig. 1, in DP-FL, excessive noise added to the gradient can lead to a significant degradation in the global model accuracy. Additionally, in DP-FL, local gradients become excessively sparse in certain training rounds. Especially when training with low privacy budgets, gradient sparsification can give rise to an abundance of noise in the uploaded gradients. This will also lead to a decrease in global model accuracy [3]. Therefore, to the best of our knowledge, we found that the crucial control over the addition of noise is still a gap that needs to be filled.

Motivated by these aforementioned issues, we aim to improve the model accuracy and availability while ensuring privacy protection by balancing privacy and accuracy of DP-FL. Exploring a fine-grained balance between privacy and accuracy has long been a critical topic of DP-FL [4], [5]. We find that most researchers aim to balance privacy protection and model accuracy in two primary ways.

1) *Gradient Sparsification*: During model training, sparsification can simplify the model complexity and reduce the risk of privacy disclosure by zeroing certain parameters [6]. Typically, these parameters pertain to less sensitive information, and setting them to zero has a minimal effect on the model performance. Weng et al. [7] proposed an FL framework that improved model accuracy by implementing sparse gradients and momentum gradient descent on both the server and client sides. SDGM [8], a sparse differential Gaussian-masking distributed SGD approach, combines sparsification techniques with Gaussian perturbation to ensure privacy guarantees within centralized stochastic gradient descent algorithms. However,

Received 25 December 2023; revised 15 May 2024; accepted 29 September 2024. Date of publication 10 October 2024; date of current version 12 December 2024. This work was supported in part by the Key Research and Development Program of China under Grant 2022YFC3005401; in part by the Key Research and Development Program of China, Yunnan Province under Grant 202203AA080009 and Grant 202202AF080003; and in part by the Science Technology Achievement Transformation of Jiangsu Province under Grant BA2021002. Recommended for acceptance by R. Sriram. (Corresponding author: Yingchi Mao.)

Benteng Zhang and Yingchi Mao are with the College of Computer Science and Software Engineering, Hohai University, Nanjing 211100, China (e-mail: yingchimao@hhu.edu.cn; 230407040003@hhu.edu.cn).

Xiaoming He is with the College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: hexiaoming@njupt.edu.cn).

Huawei Huang is with the School of Software Engineering, Sun Yat-Sen University, Zhuhai 519082, China (e-mail: huanghw28@mail.sysu.edu.cn).

Jie Wu is with the Center for Networked Computing, Temple University, Philadelphia, PA 19122 USA (e-mail: jiewu@temple.edu).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TC.2024.3477971>, provided by the authors.

Digital Object Identifier 10.1109/TC.2024.3477971

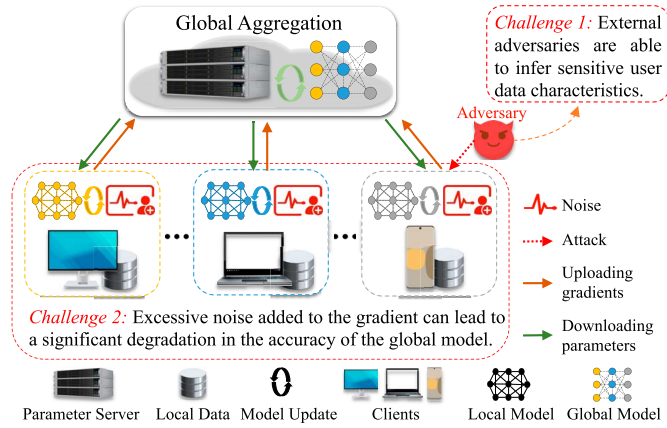


Fig. 1. Privacy leakage by MIAs in federated learning.

when training with low privacy budgets, these methods may introduce excessive noise into the uploaded gradients, leading to a significant degradation in global model accuracy.

2) *Protecting Significant Gradients*: Significant gradients typically contain highly sensitive information about model parameters. Choosing to protect significant gradients instead of all model parameters can reduce privacy-related overhead and the influence of noise on model accuracy. Previous work [9] determined that most gradient values for client updates are small, close to zero. Consequently, clients should only protect significant gradients (values far from zero) to mitigate privacy budget consumption. DP-FedSNLC [10] ascertains the significance of a gradient by evaluating alterations in the loss function and only introduces noise perturbations into significant gradients. Nevertheless, DP-FedSNLC has strong privacy protection in the early stages of global model training but slower model updates and reduced privacy protection in the later stages. In conclusion, finding a new way to precisely capture the balance between privacy and accuracy by protecting significant gradients is a challenge in DP-FL.

Given the state-of-the-art studies and the challenges described above, we propose Differential Privacy Federated Aggregation based on Significant Gradient Protection (DP-FedASGP), by integrating the idea of gradient sparsification and significant gradients protection (SGP). DP-FedASGP can prevent excessive noise addition by only protecting significant gradients and accelerate global model convergence by calculating dynamic aggregation weights for the gradients. Thus, DP-FedASGP can effectively balance the privacy and model accuracy of DP-FL, particularly under low privacy budgets.

The **contributions** of our paper are depicted as follows.

- **Originality.** We prove that introducing Laplace noise into partial significant gradients can successfully satisfy the definition of ϵ -DP. To mitigate excessive noise addition, we propose a threshold calculation method to evaluate and protect significant gradients.
- **Methodology.** To expedite the convergence of the global model, we propose a dynamic gradient aggregation

method that can dynamically calculate gradient weights and aggregate global gradients.

- **Effectiveness.** We prove the privacy guarantee and convergence of DP-FedASGP. Experiment results demonstrate that DP-FedASGP can effectively improve the accuracy and availability of the global model while ensuring the privacy protection of DP-FL.

The remainder of this paper is organized as follows. Section II presents the related work. The proposed framework is shown in Section III. The design details of DP-FedASGP are discussed in Section IV. The experiments and analysis are given in Section V. Finally, Section VI draws the conclusion.

II. RELATED WORK

A. Privacy-Preserving FL

SAFARI (sparsity-aware FL framework) [11] is designed to improve communication efficiency and reduce biases. SAFARI leverages the similarities among client models to correct and compensate for biases caused by unreliable communication. FedDST (federated dynamic sparse training) [12] focuses on dynamically extracting and training sparse subnetworks from the global network target. This approach allows each client to efficiently train its unique sparse network, reducing the need to transmit the complete model between devices and the cloud. Dai et al. [13] utilized a decentralized point-to-point communication protocol to propose Dis-PFL. Building upon the premise of gradient sparsification, DP-SIGNSGD [14] is proposed based on the concept of gradient sparsification to tackle privacy concerns in the SIGNSGD (the sign of each coordinate of the stochastic gradient vector) algorithm. To enhance privacy guarantees, FedSMP (sparsified model perturbation) [38] can sparse the local model on each client before adding noise perturbation.

However, in DP-FL, local gradients become excessively sparse in certain training rounds. When training with low privacy budgets, the aforementioned methods may introduce excessive noise into the gradient, leading to a decline in the accuracy of the global model. If the noise is reduced too much, it may not achieve the target level of privacy protection. Therefore, how to reasonably adjust the gradient noise addition to balance privacy protection and model accuracy has become an important challenge in DP-FL.

B. Privacy-Accuracy Trade-Off

Hu et al. [31] proposed Adp-PPFL (adaptive privacy-preserving FL), where the server allocates a privacy budget to each client. Clients adjust the clipping threshold for gradient clipping based on the allocated privacy budget and training rounds. However, the privacy budget refers to users' tolerance for privacy leakage, and most users participating in FL have their own privacy budgets. Adp-PPFL allocates privacy budgets to users from the server, which may lead to discrepancies between the allocated privacy budget and users' actual situations, thereby the level of privacy protection may not meet users' expectations effectively. To share specific parameters

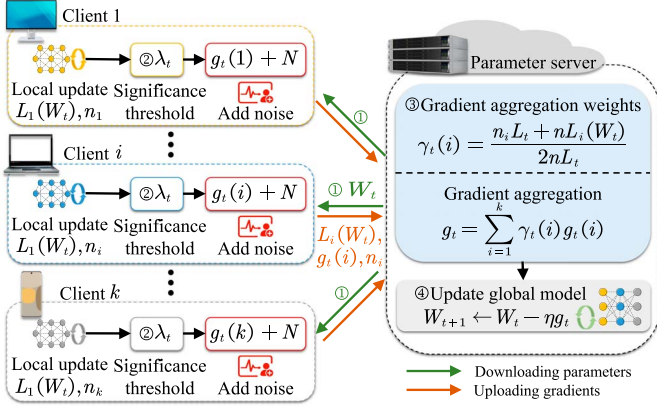


Fig. 2. The overview of DP-FedASGP. ① Global model delivery. ② Calculate the significance threshold. ③ Gradient aggregation. ④ Update global model.

from local gradients selectively, Zhao et al. [15] introduced Gaussian noise before sharing, required the determination of the amount of noise, and controlled privacy leakage through parallel training. By introducing a proxy mediation between the client and the server, the server cannot distinguish which client received the gradient. Wang et al. [16] illustrated that alterations in gradients served as a pivotal metric for gauging the susceptibility of training data to information leakage risks, and they introduced a defense strategy accordingly. By introducing gradient perturbations aligned with information leakage risks, this approach can reduce defense expenditures while upholding privacy protection. CEEP-FL (communication efficiency with enhanced privacy FL) [17] applies a filtering mechanism. This mechanism involves uploading only the significant gradients. To elevate model accuracy and preserve the privacy of distinct owner datasets, FDPBoost (federated differential privacy gradient boosting decision tree) [18] is proposed. This approach identifies sensitive features based on secure feature set indicators and assigns significant weights to protect leaf node values using the Laplace mechanism.

Nevertheless, the above methods require complex cryptographic techniques or mathematical mechanisms, which may lead to increased computational and communication costs. Besides, cross-device communication between clients, servers, and proxy devices not only increases system complexity and latency but also introduces additional privacy and security risks. Therefore, there is an urgent need to design a reasonable and effective gradient significance evaluation method to protect partial significant gradients, thus we can save privacy budgets and improve model accuracy.

III. PROPOSED FRAMEWORK

A. System Model

As shown in Fig. 2, the system model consists of a parameter server and a set of clients $\mathcal{K} = \{1, 2, \dots, i, \dots, k\}$. Client i ($\forall i \in \mathcal{K}$) has its local privacy dataset D_i and collectively trains a global model with parameters W while ensuring its protection with DP. Client i iterates locally for E times to update its

TABLE I
LIST OF MAIN SYMBOLIC PARAMETERS

Symbol	Symbol Meaning
\mathcal{K}	Client Set
k	Total number of clients
σ	Noise standard deviation
ε	Privacy budget
d	Dimensions of global model
η	Learning rate
ζ	Noise level
q	Scale parameter of noise distribution
ω	Gradient selection coefficient
δ	Relaxation term of noise
n_i	Local data size of client i
λ_t	Perturbation threshold
g_t	Global gradient in t -th training round
N	Noise
T	Training rounds
E	Local iterations
B	Local batch size
C	Fixed clipping threshold
W	Global model parameters
D_i	Local privacy dataset of client i
L_t	Global model training loss of clients
ΔS	Global sensitivity
$\gamma_t(i)$	Aggregation weight of client i
α, β	Noise for evaluating query results
D, D'	Sibling datasets
$L_i(W_t)$	Loss function for client i

local model M_i and introduces noise N into the significant gradients. Subsequently, client i uploads the processed gradient $g_t(i)$, local model training loss $L_i(W_t)$ and local data size n_i to the server. The server computes the gradient aggregation weight $\gamma_t(i)$ and aggregates the global gradient g_t by considering $g_t(i)$ and $\gamma_t(i)$. After that, the parameter server updates the global model parameters W_{t+1} . These above steps are iterated until the global model converges and attains the desired performance. The main symbolic parameters are shown in Table I.

After T training rounds, the noise introduced into the gradients will be scaled to N . With low privacy budgets, it will lead to higher noise level σ . In each training round, gradients exceeding the 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

$$g_t(i) = \begin{cases} g'_t(i) + N & \text{if } g'_t(i) + \alpha \geq \lambda_t + \beta \\ g'_t(i) & \text{otherwise} \end{cases}, \quad (1)$$

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

where $g_t(i)$ is the gradient uploaded by the client i in the t -th training round, $g'_t(i)$ 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. However, when the local data of clients are equal and non-I.I.D. (non identical and independent distribution), perturbing partial gradients and aggregating the global gradients based on FedAvg [19] will slow down the convergence of the global model. To expedite the parameter

server in aggregating the global gradients, we propose a novel method detailed as

$$g_t = \sum_{i=1}^k \gamma_t(i) g_t(i), \quad (3)$$

where g_t is the global aggregated gradient, $\gamma_t(i) \in [0, 1]$ is the weight of $g_t(i)$, and $\sum \gamma_t(i) = 1$. Section IV-C will expose the $g_t(i)$ perturbation method and the $\gamma_t(i)$ calculation method.

B. Local ϵ -Differential Privacy

The definition of DP can be associated with a privacy budget ϵ (a non-negative real number). A smaller ϵ indicates a higher level of privacy protection that users require. The definition of ϵ -DP [37] can ensure the privacy leakage caused by randomness or noise in a single query will not exceed the threshold of ϵ . 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-FedASGP satisfies ϵ -DP, i.e., to reevaluate its compliance with the following definition

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

where \mathcal{M} is the noise algorithm, e is the base of the natural logarithm, $\Pr[\cdot]$ is the probability, $\mathcal{M}(\mathbf{D}) = (x_1, \dots, x_{wd}, \dots, x_d)^T$, $\mathcal{M}(\mathbf{D}') = (x_1 + \Delta x_1, \dots, x_{wd} + \Delta x_{wd}, \dots, x_d)^T$, and O is the output vector. For any query result O , DP can guarantee that the probability ratio of generating this result with the current privacy mechanism on sibling datasets \mathbf{D} and \mathbf{D}' will not exceed e^ϵ . Section IV-A will give the privacy guarantee of DP-FedASGP.

C. Threat Model

Similar to the previous works [29], [30], we assume that all clients participating in FL training are honest-but-curious and the parameter server is honest and trustworthy. Our threat model assumes that external adversaries attempt to infer whether each sample in the given input dataset (target dataset) belongs to the training set of the client model (target model). Therefore, we choose MIAs to evaluate the privacy protection performance of DP-FedASGP. Adversaries first access the client model interface and submit a series of query requests. Subsequently, adversaries collect the model's responses to the query requests and utilize the collected model responses to infer whether specific membership identities or data features are contained within the response results through analysis methods and algorithms. Adversaries aim to determine whether a query record belongs to the training dataset of the target model. To ensure an equal number of members and non-members, we use equal-sized sets to maximize the uncertainty of the inference.

IV. GRADIENT PERTURBATION AND AGGREGATION

A. Privacy Analysis

To offer more rigorous and improved privacy protection and to facilitate the combined use of various DP mechanisms, we

choose *Laplace* noise [20] as the perturbation source. The *Laplace* noise can satisfy the ϵ -DP definition. Compared with *Gaussian* noise [21], *Laplace* noise can provide more stringent privacy safeguards at the expense of compromising information accuracy. However, DP-FedASGP will change the definition conditions of *Laplace* noise for DP. To provide security proof, this section will start with the definition of ϵ -DP and discuss how DP-FedASGP satisfies *Laplace*-based DP for partial gradients. We aim to prove that introducing *Laplace* noise into partial gradients can satisfy the requirements of ϵ -DP definition. We give Definitions 1 and 2.

Definition 1: The general definition of DP is: Given a pair of sibling datasets \mathbf{D} and \mathbf{D}' , for a function $F_{model} : \mathbf{D} \rightarrow \mathbb{R}^d$ that represents the mapping relationship from dataset \mathbf{D} to a d -dimensional space, it has a sensitivity ΔS .

Definition 2: The probability density function of the Laplace distribution for the random variable x is defined as

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

where μ is the location of the introduced noise, while the variance is $2b^2$.

Suppose that Laplace-distributed noise $Laplace_d(\frac{\Delta S}{\epsilon})$ can satisfy the ϵ -DP definition. For any domain function F_{model} with input X , the formal representation after introducing noise is given by

$$F_{model}(X) + Laplace_d\left(\frac{\Delta S}{\epsilon}\right), \quad (6)$$

where $\frac{\Delta S}{\epsilon}$ is the scale parameter of the Laplace distribution. In order to simplify the proof process without affecting the generalization of the results, we give Assumptions 1 and 2.

Assumption 1: For any domain function F_{model} with input dataset \mathbf{D} , F_{model} is given by

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

Assumption 2: For any random variable x_i in dataset \mathbf{D} , $x_i = 0$.

Using these conditions, we can make our proof without losing generality. We give Theorem 1.

Theorem 1 (Privacy Guarantee of DP-FedASGP): If (6) is feasible and Assumptions 1, 2 hold, then introducing Laplace noise into partial significant gradients can satisfy the definition of ϵ -DP and ensure the privacy of gradients.

Proof: Please refer to Appendix A. \square

According to Theorem 1, it can be concluded that introducing Laplace noise into partial significant gradients can ensure the privacy of the gradients. Therefore, DP-FedASGP does not affect the convergence of the global model.

B. Gradient Perturbation Mechanism

We aim to provide stricter and more robust privacy protection while facilitating the joint use of multiple DP mechanisms. As shown in Fig. 3, for client i , after computing the local λ_t , α , and β , noise is introduced into the query results that exceed λ_t in d queries. We combine DP with Laplace noise, referred to as (ϵ, δ) -DP. When the relaxation term $\delta = 0$, the

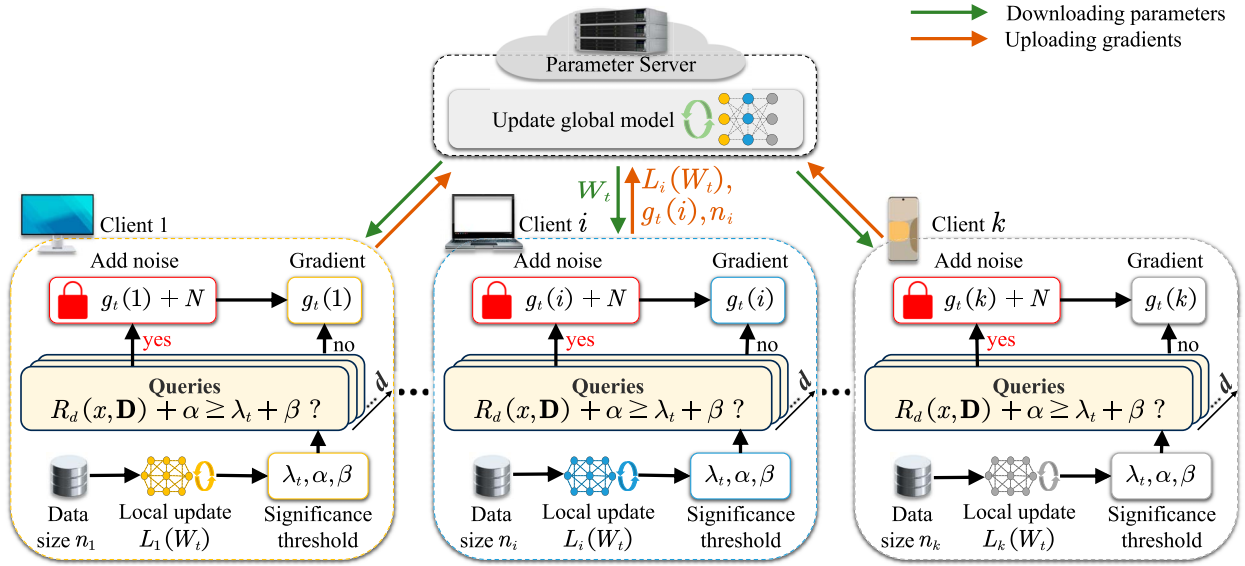


Fig. 3. Gradient perturbation of each client.

random algorithm \mathcal{M} can satisfy the ε -DP definition. We give Definitions 3 and 4.

Definition 3: Given a random algorithm \mathcal{M} and input datasets \mathbf{D}, \mathbf{D}' , the formal definition of ε -DP is defined as

$$\Pr[\mathcal{M}(\mathbf{D}) \in S] \leq e^\varepsilon \Pr[\mathcal{M}(\mathbf{D}') \in S] + \delta, \quad (8)$$

where $S \subseteq \text{Range}(\mathcal{M})$ and $\delta = 0$.

Definition 4: Given sequentially executed random algorithms \mathcal{M}_1 and \mathcal{M}_2 satisfy ε_1 -DP and ε_2 -DP, respectively. $\mathcal{M}_1, \mathcal{M}_2$ satisfy $(\varepsilon_1 + \varepsilon_2)$ -DP, which is defined as

$$\Pr[\mathcal{M}(\mathbf{D})] \leq e^{\varepsilon_1 + \varepsilon_2} \Pr[\mathcal{M}(\mathbf{D}')]. \quad (9)$$

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

$$Q = R(x, \mathbf{D}) + \mathbf{N}. \quad (10)$$

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

$$\Pr(\mathbf{N}) = \frac{\varepsilon}{2\Delta S} e^{-\frac{\varepsilon}{\Delta S} |\mathbf{N}|}. \quad (11)$$

According to the composition theorem of DP mechanisms that satisfy the Laplace distribution in FL, the simultaneous execution of multiple queries will result in a linear increase in the consumed privacy budget. Let d denote the dimension of the FL model and x is the input parameters of the d queries, the update of d parameters by a single client is equivalent to answering d queries concurrently. 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

$$Q(x, \varepsilon) = R(x, \mathbf{D}) + \mathbf{N}(\varepsilon). \quad (12)$$

To reduce the privacy budget consumption of simultaneously executing multiple queries, we introduce the idea of sparse vectors [22]. Laplace noise is only introduced 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 introduced only when $R_d(x, \mathbf{D}) + \alpha \geq \lambda + \beta$, then we have

$$A_d = R_d(x, \mathbf{D}) + N_d, \quad (13)$$

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, and q is the scale parameter of Laplacian noise distribution. α and β are additional noise that evaluate the importance of the query result, which follow the Laplace noise distributions $\text{Lap}(q\Delta S/\varepsilon_1)$ and $\text{Lap}(q\Delta S/\varepsilon_2)$, respectively. N_d is the noise used to perturb the query result, which follows the Laplace noise distribution $\text{Lap}(q\Delta S/\varepsilon_3)$.

However, the premise of using the above gradient perturbation method is that the total privacy budget ε satisfies $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$. Therefore, we need to prove Theorem 2.

Theorem 2: If (13) is feasible, then the proposed DP-FedASGP can introduce Laplace noise into partial significant gradients, and the total privacy budget $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$.

Proof: Please refer to Appendix B. \square

Due to 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. 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 [23] method into the selection of λ . We set different thresholds λ 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 accelerate the model convergence. In the later training rounds, when the parameters tend to stabilize and there is less gradient information, a smaller threshold λ is set to save the privacy budget. The calculation of the threshold λ_t is given by

$$\lambda_t = \min \left(\text{sort}(g') \left[\left\lceil \frac{t|W|}{T} \right\rceil \right], \text{sort}(g') \left[\left\lceil \frac{9|W|}{10} \right\rceil \right] \right), \quad (14)$$

where $|W|$ is the total number of model parameters, $\text{sort}(\cdot)$ is the sorting result in ascending order, and g' is the locally clipped gradient.

C. Gradient Aggregation Mechanism

The most commonly referenced algorithm in FL is the FedAvg algorithm, in which the weights of the gradients are typically fixed and determined based on the size of local training data. After T training rounds, the global model objective is

$$\min \sum_{i=1}^k \frac{n_i}{n} L_i(W), \quad (15)$$

where $L_i(W)$ is the loss function used to train the local model of client i , $n = \sum_{i=1}^k n_i$ is the total data size, and n_i is the local data size of client i . The impact of local loss on the global objective depends entirely on the local data size. We hope that the global loss function can truly reflect the aggregated global model by gradient aggregation weights on the server-side. However, the accuracy of the gradients decreases after noise is introduced, especially when partial gradients are perturbed. The information contained in the gradients uploaded by each client may be completely different from the previous values.

As shown in Fig. 2, based on the effectiveness of local training, we dynamically calculate the gradient aggregation weight γ in each training round. Let $L_i(W_t)$ denote the local model training loss for client i in the t -th training round, the total model training loss for client i is $L_t = \sum_{i=1}^k L_i(W_t)$. 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 $\gamma_t(i)$ for client i is given by

$$\gamma_t(i) = \frac{n_i L_t + n L_i(W_t)}{2n L_t}, \quad (16)$$

where $\sum_{i=1}^k \frac{n_i}{n} = 1 \cap \sum_{i=1}^k \frac{L_i(W_t)}{L_t} = 1$ always holds, and the gradient aggregation weight $|\gamma_t| = \sum_{i=1}^k \gamma_t(i) = 1$ for each client in each training round is always true.

D. Convergence Analysis

In this section, we prove the convergence of the DP-FedASGP within the FL framework. We give Assumptions 3, 4 and Definition 5.

Assumption 3 (Lipschitz Smoothness): We assume that the loss function $L(\cdot)$ is differentiable, and each client's local loss function $\nabla L(\cdot)$ is l -Lipschitz continuous, i.e., $\forall i \in \{1, 2, \dots, k\}$

$$\|\nabla L_i(\mathbf{W}) - \nabla L_i(\mathbf{W}')\| \leq l \|\mathbf{W} - \mathbf{W}'\|. \quad (17)$$

Assumption 4 (Bounded Gradient): By the nature of gradient descent, we assume that each client's local gradient is bounded. Therefore, there is a constant M_i such that for any training round t and client i , we have

$$\|\nabla L_i(W_t)\| \leq M_i. \quad (18)$$

According to the gradient descent method, the global gradient g_t can be expressed as the gradient of the global loss function $L(\cdot)$, which is given by

$$g_t = \nabla L(W_t) = \frac{1}{k} \sum_{i=1}^k \nabla L_i(W_t) \gamma_t(i). \quad (19)$$

Definition 5: The update rule of the global model parameter W after the training round t is defined as

$$W_{t+1} \leftarrow W_t - \eta g_t, \quad (20)$$

where η is the learning rate and g_t is the global gradient of training round t .

In DP-FedASGP, a certain degree of *Laplacian* noise is added to each client's gradient $g_t(i)$ to protect the true gradient and satisfy the ε -DP. At the same time, to prevent the instability of the global model parameter update caused by too large gradients, the gradient clipping function can limit the gradient to a certain range. To prove the convergence of DP-FedASGP, we need to prove Theorem 3.

Theorem 3 (Convergence Guarantee of DP-FedASGP): If (19) is feasible and Assumptions 3, 4 hold, then the global gradient g_t after gradient clipping in each training round is bounded and the global model parameters W_t can converge to a finite value.

Proof: Please refer to Appendix C. \square

E. Algorithm Design

The gradient perturbation and aggregation process of DP-FedASGP can be divided into three key stages:

1) *Construct A Gradient Perturbation Method Based on Sparse Vectors:* Not every gradient from clients holds equal significance. In each training round, we begin by computing the threshold λ_t to evaluate gradient significance. Subsequently, employing decision criteria $R_d(x, \mathbf{D}) + \alpha \geq \lambda_t + \beta$, we only introduce noise into significant gradients $g_t(i)$. The total privacy budget essential for global model training, denoted as $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$, determines the gradient information that each client ultimately needs to upload.

2) *Build Dynamic Aggregation Weights Calculation Method:* In each training round, each client possesses a local model training loss function $L_i(W_t)$. Therefore, we can calculate the total model training loss L_t for all clients. The gradient aggregation weight $\gamma_t(i)$ in the current training round is determined based on the data distribution.

3) *Formulate The Global Model Gradient Aggregation Method:* For the parameter server, based on the perturbed gradients uploaded by each client obtained in stage 1 and the gradient aggregation weights obtained in stage 2, the final global model aggregation is accomplished.

Algorithm 1: DP-FedASGP

Input: k clients, D_i , B , E , T , C , ΔS , ε_1 , ε_2 , η , σ
Output: Global model parameters W

```

1 Initialize  $W$ ,  $W_t$ 
2 for each training round  $t \in T$  do
3   for each client  $i$  in parallel do
4      $W \leftarrow W_t$ 
5      $g_i, L_i, n_i \leftarrow \text{clientTrain}(W, D_i, T, t)$ 
6   end
7    $L \leftarrow \sum L_i, n \leftarrow \sum n_i$ 
8    $\gamma_i \leftarrow \frac{n_i L + n L_i}{2nL}$ 
9    $g \leftarrow \sum \gamma_i g_i$ 
10   $W_{t+1} \leftarrow W_t - \eta g$ 
11 end
12 function  $\text{clientTrain}(W, D_i, T, t)$ 
13 begin
14    $n \leftarrow D$ 
15   for each local epoch in  $E$  do
16      $g_e \leftarrow \nabla L(W)$ 
17      $W \leftarrow W - \eta g_e$ 
18   end
19    $g \leftarrow \sum g_e, L = \sum \Delta M(W)_e$ 
20    $g' \leftarrow \frac{g}{\max(1, \|g\|_2)}$ 
21    $\alpha \leftarrow \text{Lap}(\frac{\Delta S}{\varepsilon_1}), \beta \leftarrow \text{Lap}(\frac{\Delta S}{\varepsilon_2})$ 
22    $\lambda_t = \min(\text{sort}(g')[\lceil \frac{t|W|}{T} \rceil], \text{sort}(g')[\lceil \frac{9|W|}{10} \rceil])$ 
23   if  $g' + \alpha \geq \lambda + \beta$  then
24      $g \leftarrow g' + N$ 
25   else
26      $g \leftarrow g'$ 
27   end
28   return  $g, L, n$ 
29 end

```

The details are shown in **Algorithm 1**. DP-FedASGP comprises the following steps: *a*) Initialize the server and client models (line 1). *b*) In each communication round, client i iterates local batch model training on their local privacy data, computing gradients, and local loss (lines 15-18). *c*) Client i computes the sparse inequality noise levels α and β (lines 19-21). *d*) Client i calculates the noise perturbation threshold λ and introduces noise into significant gradients based on the sparse inequality (lines 22-28). *e*) Each client uploads the perturbed gradients g , local loss L , and local data size n (line 5). *f*) The server calculates the total training loss and total data size and computes each gradient aggregation weight γ_i (lines 7-8). *g*) The server aggregates the perturbed gradients uploaded by clients based on the gradient aggregation weights (line 9). *h*) The parameter server updates a new global model and sends the new model to each client (line 10).

F. Complexity Analysis

In DP-FedASGP, each client performs local training and updates its local model for E times, resulting in an algorithmic complexity denoted as $\mathcal{O}(\sum_{e=1}^E |\nabla|_e) = \mathcal{O}(E)$. In the

FL framework with k clients participating in training, the algorithmic complexity for computing local gradients is $\mathcal{O}(k)$. As the clients are training models in parallel, the algorithmic complexity is transformed into $\mathcal{O}(1)$. Assuming the parameter server performs T training rounds, the overall complexity of DP-FedASGP is expressed as $\mathcal{O}(ET)$. Since E is much smaller than T , i.e., $E \ll T$, the overall complexity of the DP-FedASGP algorithm is $\mathcal{O}(n)$.

V. PERFORMANCE EVALUATION**A. Experimental Settings**

Experimental Environment. The experiments are conducted with a parameter server and a group of clients participating in FL training. PyTorch is used as the deep learning framework, and the Python version employed is 3.6. The computing nodes run on the 64-bit Ubuntu 20.04 LTS operating system, with a CUDA driver version of 11.0. The CPU used is an Intel(R) Xeon(R) Gold 6326 @2.90GHz, equipped with 256GB of RAM, 4TB of hard disk, and an NVIDIA A100 GPU with 80GB of GPU memory.

Datasets and Target Models. We employ 3 image datasets, 1 text dataset, and 3 target models. We use the Dirichlet function $\text{Dir}(\varphi = 1)$ [39] to divide the datasets for clients to generate non-independent and identically distributed (non-IID) training datasets. Note that the higher the φ value, the more similar the distribution of the training dataset is allocated among clients. Res50 represents ResNet-50 in Tables III, IV, and V.

- **MNIST** [34] dataset consists of 60,000 labeled training images and 10,000 labeled test images. The data comprises hand-written digit images representing all digits from 0 to 9, with a fixed size of 28×28 pixels in grayscale. We use a convolutional neural network (CNN) model and a Residual Network (ResNet-50) for image classification tasks. CNN consists of 2 convolutional layers (5×5 and ReLU activation, each followed by 2×2 max pooling), 2 fully connected layers, and Softmax normalizes the final output. ResNet-50 uses the same default settings as the 50-layer architecture in [36].
- **CIFAR-10** [35] and **CIFAR-100** [35] datasets both consist of 50,000 labeled training images and 10,000 labeled test images. These images belong to 10 and 100 categories respectively, with each category representing one of them. The images are fixed-sized color images of 32×32 pixels. We also use CNN and ResNet-50 for image classification tasks. CNN consists of 2 convolutional layers (5×5 and ReLU activation, each followed by 2×2 max pooling), 3 fully connected layers, and Softmax normalizes the final output. ResNet-50 is set up the same way as MNIST.
- **Shakespeare** [19] dataset is built from *The Complete Works of William Shakespeare*. Similar to [33], each client has one or more lines for training or testing. We train a recurrent neural network (RNN) model for predicting the next character. RNN takes a sequence of 80 characters as input and consists of an embedding layer (80×8), two LSTM layers (80×256), and a dense layer (80×90).

Baselines. We consider the following comparative methods.

TABLE II
DATASETS DETAILS AND HYPERPARAMETER SETTINGS

Datasets	MNIST	CIFAR-10	CIFAR-100	Shakespeare
Type	Image	Image	Image	Text
Models	CNN	CNN	CNN	RNN
	Resnet50	Resnet50	Resnet50	
Clients	100	100	100	715
Train Size	60,000	50,000	50,000	16,068
Test Size	10,000	10,000	10,000	2,356
Batch Size	128	128	128	4
Training Round	200	500	500	1,000
Learning Rate	0.1	0.05	0.05	1

- DP-FedAvg [24] was once the state-of-the-art DP variant of the FedAvg algorithm. It implements client-level DP, where each client uses a clipping threshold C for gradient clipping, followed by adding noise N .
- cpSGD [25] combines gradient quantization and DP.
- DP-FedSNLC [10] introduces sparse noise into gradients based on clipping losses and privacy budget costs.
- FedSMP- top_k [38] is currently the state-of-the-art client-level DP method for balancing accuracy and privacy through sparsification. DP-FedSMP simplifies the local model updates of clients by retaining only important coordinate subsets and then adds noise to perturb the retained coordinate values.

Hyperparameter Settings. SGD algorithm is used for local gradient computation. The clipping threshold $C = 1$ and the privacy budgets $\varepsilon = \{0.1, 0.2, 0.5, 1, 2, 4\}$. We randomly select 10% of the clients in each training round to participate in the FL training. The details are shown in Table II.

Attack models. We consider the following threat models.

- *Basic-MIA* [32]: Threshold-based MIA. Adversaries compute the prediction confidence of the target model on a shadow dataset, then select a confidence threshold that achieves the highest attack accuracy on the shadow dataset. If the confidence of a queried record exceeds this threshold, adversaries will infer the member record.
- *ML-Leaks (Adversary 1)* [27]: Adversaries divide the shadow dataset D_{shadow} into training datasets D_{shadow}^{train} and testing datasets D_{shadow}^{test} , and then train a shadow model M_{shadow} based on the data from D_{shadow}^{train} . For each record in D_{shadow} , adversaries select the three largest posterior values from the output of M_{shadow} and label them as 1 or 0 to train the attack model. Finally, adversaries feed the three largest posterior values into the attack model to obtain predictions about membership.
- *White-box* [28]: Adversaries train attack models on both the training and testing datasets to learn the differences in member inference. Adversaries process multiple observed target model inputs simultaneously, capturing the correlations between parameters in each training round.
- *CS-MIA* [26]: The state-of-the-art MIA. Adversaries first divide the dataset into training datasets D_{train} and testing datasets D_{test} , and incorporate D_{train} into FL. Then, adversaries compute the confidence series of the shadow model on D_{train} and D_{test} , constructing a labeled confidence series set used for training the attack model. In the

inference phase, for a given target record d_{target} , adversaries compute the confidence series of the target model on d_{target} as the input to the pre-trained attack model, finally determining the membership of d_{target} .

Metrics. Various metrics of the experiments provide an intuitive description of the model's training performance.

- *Privacy Protection:* The experiments are designed to assess the privacy protection performance of the five methods. A lower accuracy of inference attacks in the experimental results indicates better privacy protection.
- *Global Model Availability:* The experimental results indicate that higher global test accuracy corresponds to higher model training accuracy and availability. When training with low privacy budgets, we should pay special attention to the changes in the average global test accuracy during model training.
- *Applicability of DP-FedASGP:* The experiments compare the global test accuracy of the trained model to evaluate the applicability of DP-FedASGP under higher privacy budgets. Note that a higher average accuracy in the experimental results indicates better applicability.

B. Privacy Protection

The most direct and intuitive way to evaluate the performance of privacy protection is by incorporating inference attack methods during the model training. Since the privacy budget affects the overall accuracy of inference attacks, our experiments choose the MNIST, CIFAR-10/100 datasets and $\varepsilon = \{0.1, 0.2, 0.5\}$, using Basic-MIA, ML-Leaks, White-box, and CS-MIA as attack methods for comparative experiments. We will analyze the privacy protection performance of DP-FedASGP and other methods. Note that a lower accuracy of inference attacks in the experimental results indicates better privacy protection. The results are shown in Table III.

1) *Training with Different Privacy Budgets and Different Model Training Methods:* The Basic-MIA attacks perform the worst, while the CS-MIA attacks perform the best. Given that Basic-MIA is much simpler than the other inference attack methods, the attack accuracy of Basic-MIA is the lowest across all five training methods. Since CS-MIA attacks can access more information on training and testing data, the effectiveness of CS-MIA is better than that of ML-Leaks and White-box.

2) *Training with Different Privacy Budgets and Different Inference Attack Methods:* As shown in Table III, cpSGD has the lowest attack accuracy, so cpSGD has the best privacy protection performance. This is because cpSGD combines gradient quantization and DP, which can ensure privacy protection definitions. However, quantization does not scale the amount of gradients. Instead, quantization adds some privacy protections and makes inference attacks more challenging. Therefore, cpSGD offers better privacy protection compared to DP-FedASGP. DP-FedAvg has lower attack accuracy than DP-FedSNLC, FedSMP, and DP-FedASGP. This is because DP-FedAvg only prevents

TABLE III
ATTACK ACCURACY OF DIFFERENT ATTACK METHODS WITH DIFFERENT TRAINING METHODS

Privacy Budget	Model	Method (DP-)	MNIST				CIFAR-10				CIFAR-100			
			Basic-MIA	ML-Leaks	White-box	CS-MIA	Basic-MIA	ML-Leaks	White-box	CS-MIA	Basic-MIA	ML-Leaks	White-box	CS-MIA
0.1	CNN	FedAvg [24]	50.01%	50.02%	50.02%	51.09%	50.86%	51.42%	53.61%	65.46%	51.45%	53.42%	55.01%	65.72%
		cpSGD [25]	50.01%	50.01%	50.02%	51.09%	50.73%	51.31%	53.57%	65.31%	51.24%	53.31%	54.96%	65.67%
		FedSNLC [10]	50.09%	50.21%	50.24%	51.15%	51.38%	58.84%	62.59%	67.19%	54.77%	60.26%	65.86%	67.25%
		FedSMP [38]	50.06%	50.15%	50.20%	51.17%	51.04%	51.87%	54.76%	65.85%	51.95%	54.23%	55.93%	66.10%
		FedASGP	50.04%	50.09%	50.11%	51.15%	50.89%	51.80%	54.38%	65.49%	51.92%	54.18%	55.84%	65.91%
	Res50	FedAvg	50.01%	50.01%	50.01%	51.03%	50.47%	51.17%	53.07%	64.80%	51.07%	52.31%	54.26%	65.21%
		cpSGD	50.01%	50.01%	50.01%	51.02%	50.45%	51.08%	53.06%	64.67%	51.05%	52.18%	54.15%	65.19%
		FedSNLC	50.03%	50.17%	50.20%	51.11%	50.84%	57.93%	60.82%	66.59%	53.17%	59.30%	62.68%	67.04%
		FedSMP	50.03%	50.10%	50.15%	51.08%	50.63%	51.49%	53.95%	64.97%	51.46%	52.90%	55.15%	65.66%
		FedASGP	50.01%	50.05%	50.07%	51.06%	50.52%	51.32%	53.67%	64.82%	51.20%	52.92%	54.27%	65.27%
0.2	CNN	FedAvg	50.04%	50.18%	50.35%	51.28%	53.07%	55.12%	60.47%	70.91%	55.63%	58.03%	61.76%	71.21%
		cpSGD	50.03%	50.16%	50.23%	51.22%	52.99%	55.06%	60.42%	70.85%	55.49%	57.65%	61.64%	71.06%
		FedSNLC	50.15%	50.27%	50.55%	51.35%	55.56%	60.69%	67.19%	73.21%	58.46%	65.88%	69.49%	73.85%
		FedSMP	50.13%	50.25%	50.43%	51.31%	53.63%	55.39%	60.75%	71.54%	55.87%	58.98%	62.21%	71.73%
		FedASGP	50.09%	50.21%	50.36%	51.30%	53.58%	55.33%	60.70%	71.35%	55.84%	58.91%	62.07%	71.57%
	Res50	FedAvg	50.02%	50.11%	50.24%	51.19%	52.26%	54.57%	59.65%	66.94%	54.42%	55.23%	60.11%	67.26%
		cpSGD	50.02%	50.08%	50.16%	51.18%	52.21%	54.41%	59.59%	66.88%	54.34%	55.11%	60.01%	67.18%
		FedSNLC	50.07%	50.20%	50.41%	51.27%	52.55%	58.33%	65.88%	69.10%	56.11%	59.45%	67.82%	69.84%
		FedSMP	50.05%	50.18%	50.37%	51.24%	52.46%	54.68%	60.12%	67.06%	54.65%	55.40%	60.57%	67.75%
		FedASGP	50.03%	50.14%	50.28%	51.22%	52.41%	54.61%	59.92%	67.04%	54.54%	55.28%	60.45%	67.34%
0.5	CNN	FedAvg	50.13%	50.39%	50.83%	51.46%	55.24%	59.32%	65.23%	73.85%	60.29%	63.22%	68.10%	74.49%
		cpSGD	50.10%	50.36%	50.81%	51.40%	55.08%	59.25%	65.17%	73.67%	60.02%	63.07%	67.70%	74.33%
		FedSNLC	50.26%	50.52%	50.91%	51.55%	58.91%	62.84%	70.36%	78.72%	64.41%	70.15%	73.63%	78.97%
		FedSMP	50.19%	50.49%	50.88%	51.52%	55.49%	59.52%	65.68%	74.39%	60.73%	64.10%	68.41%	74.89%
		FedASGP	50.17%	50.44%	50.86%	51.48%	55.45%	59.47%	65.60%	74.11%	60.74%	64.06%	68.34%	74.65%
	Res50	FedAvg	50.06%	50.26%	50.54%	51.35%	54.15%	58.46%	63.91%	71.98%	58.52%	60.66%	65.51%	72.59%
		cpSGD	50.05%	50.21%	50.53%	51.32%	54.12%	58.22%	63.85%	71.57%	58.46%	60.49%	65.36%	72.51%
		FedSNLC	50.19%	50.44%	50.65%	51.49%	58.08%	61.51%	66.72%	75.62%	61.43%	64.33%	70.15%	75.22%
		FedSMP	50.14%	50.33%	50.59%	51.41%	54.80%	58.50%	64.10%	72.45%	58.60%	60.75%	65.97%	72.98%
		FedASGP	50.11%	50.30%	50.58%	51.34%	54.64%	58.45%	64.05%	72.42%	58.57%	60.70%	65.85%	72.86%

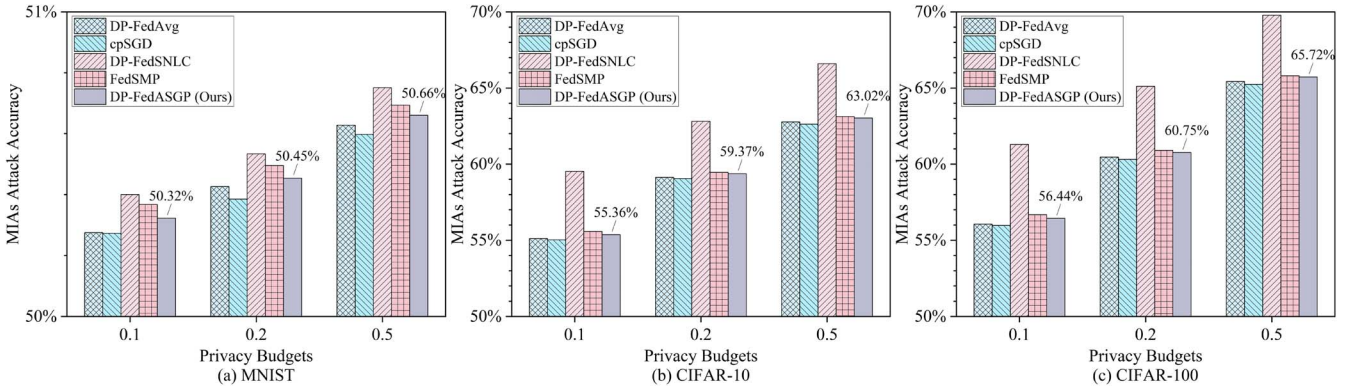


Fig. 4. Average attack accuracy with different models and privacy budgets across CNN and ResNet-50.

the addition of larger noise and utilizes a fixed clipping threshold for gradient clipping. This leads to more noise being introduced compared to DP-FedSNLC, FedSMP, and DP-FedASGP. Hence, the privacy protection of DP-FedAvg is second only to cpSGD and better than DP-FedASGP. The attack accuracy of DP-FedSNLC is much higher than the other methods, and it offers the worst privacy protection performance. DP-FedSNLC introduces noise into significant gradients based on the changes in the loss function. In the early training stages, gradients change significantly, while in the later stages, gradients gradually diminish. This leads to strong privacy protection in the early stages of global model training and weaker privacy protection in the later stages. Therefore, DP-FedSNLC performs worse against inference attacks compared to DP-FedASGP, especially under complicated MIAs. FedSMP achieves higher attack accuracy compared to DP-FedASGP because FedSMP sparsifies the model before adding noise perturbation. The degree of

model sparsification cannot adapt well to changes in the privacy budget, making it more susceptible to attacks.

DP-FedASGP has slightly higher attack accuracy than DP-FedAvg and cpSGD but is significantly lower than DP-FedSNLC and FedSMP. This indicates that the privacy protection performance of DP-FedASGP is slightly worse than DP-FedAvg and cpSGD but significantly better than DP-FedSNLC and FedSMP. This is because DP-FedASGP only introduces noise into partial significant gradients and dynamically computes the gradient significance threshold in each training round. This results in less introduced noise compared to DP-FedAvg and cpSGD but makes DP-FedASGP offer a slightly weaker defense against inference attacks compared to DP-FedAvg and cpSGD. Furthermore, we have compiled the average attack accuracy of the four attack methods across CNN and ResNet-50 under different privacy budgets and different model training methods, details are shown in Fig. 4.

TABLE IV
GLOBAL TEST ACCURACY WITH DIFFERENT PRIVACY BUDGETS FOR
DIFFERENT TRAINING METHODS

Privacy Budget	Method (DP-)	MNIST		CIFAR-10		CIFAR-100		Shakespeare RNN
		CNN	Res50	CNN	Res50	CNN	Res50	
0.1	FedAvg [24]	88.34%	89.10%	34.49%	45.73%	10.36%	26.21%	20.16%
	cpSGD [25]	85.53%	86.42%	31.92%	44.27%	8.92%	24.91%	18.90%
	FedSNLC [10]	90.74%	91.47%	39.82%	49.69%	13.94%	27.44%	22.34%
	FedSMP [38]	91.02%	92.11%	40.55%	51.06%	14.29%	27.87%	22.95%
	FedASGP	91.16%	92.53%	40.96%	51.35%	14.34%	28.05%	23.04%
0.2	FedAvg	91.68%	92.74%	42.88%	56.10%	16.17%	33.51%	27.87%
	cpSGD	89.15%	90.11%	39.51%	54.53%	14.89%	32.11%	26.32%
	FedSNLC	92.92%	94.45%	46.53%	57.31%	18.91%	34.37%	28.85%
	FedSMP	93.10%	94.59%	46.72%	57.45%	19.38%	34.50%	29.06%
	FedASGP	93.24%	94.80%	46.90%	57.65%	19.44%	34.82%	29.18%
0.5	FedAvg	93.87%	94.33%	48.36%	65.23%	20.21%	39.80%	31.46%
	cpSGD	90.63%	91.55%	45.94%	62.98%	18.31%	38.15%	30.12%
	FedSNLC	94.28%	95.40%	51.94%	66.74%	22.15%	41.40%	33.07%
	FedSMP	94.53%	95.43%	51.20%	66.81%	22.53%	41.39%	33.24%
	FedASGP	94.75%	95.82%	51.37%	66.80%	22.63%	41.57%	33.41%

3) *Training with Different Datasets and Different Privacy Budgets*: As shown in Fig. 4, DP-FedASGP exhibits slightly higher average attack accuracy than DP-FedAvg and cpSGD but significantly lower than DP-FedSNLC and FedSMP, especially when training models with high-complexity datasets.

4) *Summary*: DP-FedASGP is effective in defending against inference attacks under low privacy budgets. The privacy protection performance of DP-FedASGP is similar to DP-FedAvg and cpSGD but significantly better than DP-FedSNLC and FedSMP, especially when training with CIFAR-10/100.

C. Global Model Availability

In this section, by evaluating the availability of DP-FedSGP using the global test accuracy of the model, we will prove that DP-FedASGP can offer the best global model availability compared to the other four methods. As detailed in Table IV and illustrated in Figs. 5 and 7, we present the global test accuracy and average test accuracy results of DP-FedAvg, cpSGD, DP-FedSNLC, FedSMP, and DP-FedASGP. In addition, to more intuitively show the differences between DP-FedASGP and other methods, we also provide the average global test accuracy on four datasets in Table V. Note that a higher global test accuracy in the experimental results indicates higher model training availability.

1) *The Lower The Privacy Budget, The Better The Availability*: As shown in Table IV, when training the CNN and ResNet-50 with the MNIST dataset and privacy budget $\epsilon = 0.1$, DP-FedASGP demonstrates a substantial global test accuracy improvement of approximately 2.82% and 3.43% compared to DP-FedAvg, respectively. When $\epsilon = 0.5$, DP-FedASGP still outperforms DP-FedAvg, with a slightly reduced improvement of about 0.88% and 1.49%, respectively. Similar trends are observed during training with the CIFAR-10/100 and Shakespeare datasets, where the increase in global test accuracy with DP-FedASGP is more pronounced at $\epsilon = 0.1$ compared to 0.5. Therefore, DP-FedASGP can improve the availability of the global model with low privacy budgets.

2) *The More Complex the Dataset, the Better the Availability*: When training with the relatively simple MNIST dataset, the performance differences among these five methods are

not particularly significant. However, when training with the CIFAR-10/100 and Shakespeare datasets, which exhibit higher data complexity, DP-FedASGP stands out by achieving the highest global test accuracy. In particular, during training with complex datasets, characterized by moderate data complexity, DP-FedASGP significantly outperforms DP-FedAvg and cpSGD in terms of global test accuracy. DP-FedASGP only introduces noise perturbation into partial significant gradients in each training round and dynamically calculates the gradient aggregation weights. These improvements in DP-FedASGP can enhance the precision of gradient perturbation, which is particularly beneficial for complex datasets.

3) *The Average Global Test Accuracy*: We average the global test accuracy of $\epsilon = \{0.1, 0.2, 0.5\}$ on the four datasets, DP-FedASGP has a higher average global test accuracy than the other methods. As shown in Table V, DP-FedASGP achieves a higher average global test accuracy than DP-FedAvg, cpSGD, DP-FedSNLC, and FedSMP on the four datasets, with improvements of approximately 2.62%, 4.71%, 0.45%, and 0.19%, respectively. cpSGD combines gradient quantization and DP to ensure privacy protection definition. However, since cpSGD quantizes gradients, it is equivalent to introducing privacy noise perturbation. Therefore, under the same privacy budget setting, cpSGD has the lowest average global test accuracy. DP-FedAvg can only prevent the addition of larger noise. Thus, DP-FedAvg introduces more noise than DP-FedSNLC, FedSMP, and DP-FedASGP. DP-FedSNLC evaluates the changes in the loss function to determine whether gradients are important and then introduces noise perturbation. FedSMP tends to overly sparsify the model under low privacy budgets, leading to poor performance, but FedSMP still effectively reduces the addition of noise. Therefore, DP-FedAvg, DP-FedSNLC, and FedSMP outperform cpSGD in terms of global test accuracy.

4) *Summary*: As the MNIST dataset has relatively low complexity, these five methods exhibit similar global test accuracy, and the precision improvement can be negligible. However, for the CIFAR-10/100 and Shakespeare datasets, with a significant increase in dataset complexity, the global test accuracy of DP-FedASGP outperforms DP-FedAvg, cpSGD, DP-FedSNLC, and FedSMP. When training with low privacy budgets, the average global test accuracy of DP-FedASGP on the four datasets is higher than other methods. Therefore, DP-FedASGP can offer the best global model availability among these five methods.

D. Applicability Analysis Under Higher Privacy Budgets

In Sections V-B and V-C, the privacy protection performance of DP-FedASGP is approximately equivalent to DP-FedAvg and cpSGD. When training with low privacy budgets, DP-FedASGP can offer better global model availability than other methods. Therefore, in this section, we choose DP-FedAvg as the comparative method. We will prove that DP-FedASGP can maintain excellent applicability even under higher privacy budgets.

Table VI and Fig. 6 present the experimental results of global test accuracy for DP-FedAvg and DP-FedASGP (CNN model)

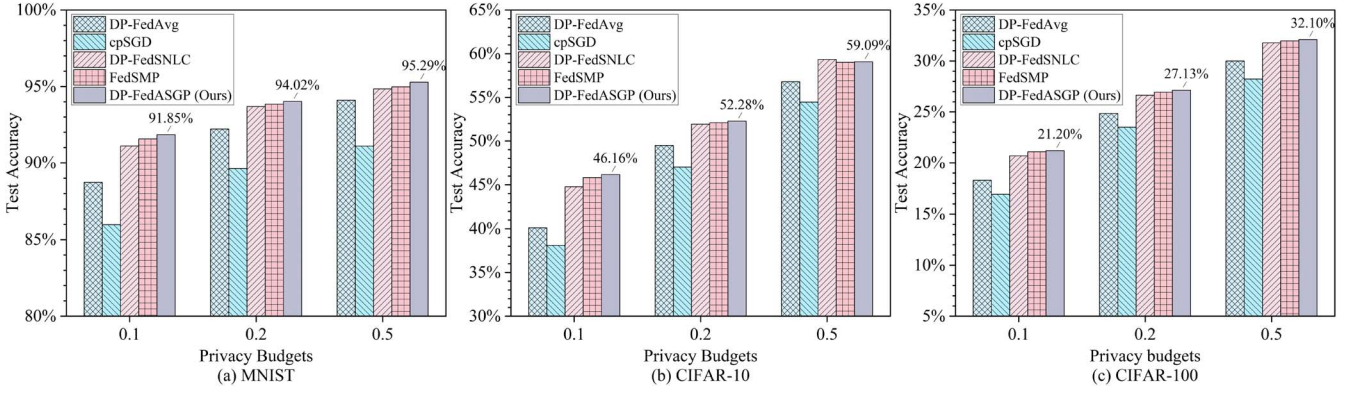


Fig. 5. Average global test accuracy with different training methods and privacy budgets across CNN and ResNet-50.

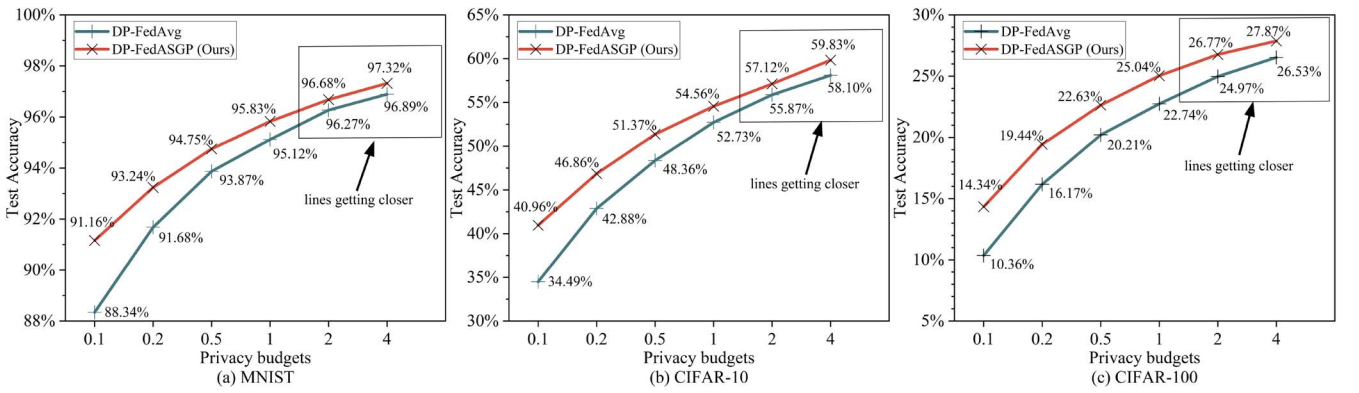


Fig. 6. Global test accuracy with higher privacy budgets on CNN.

TABLE V
AVERAGE GLOBAL TEST ACCURACY WITH DIFFERENT TRAINING METHODS ON DIFFERENT DATASETS

Dataset	Model	DP-FedASGP	DP-FedAvg [24]	cpSGD [25]	DP-FedSNLC [10]	FedSMP [38]
MNIST	CNN	93.05%	91.30% (-1.75)	88.33% (-4.72)	92.65% (-0.40)	92.88% (-0.17)
	Res50	94.38%	92.06% (-2.32)	89.36% (-5.02)	93.77% (-0.61)	94.04% (-0.34)
CIFAR-10	CNN	46.40%	41.91% (-4.49)	39.12% (-7.28)	46.10% (-0.30)	46.16% (-0.24)
	Res50	58.60%	55.69% (-2.91)	53.93% (-4.67)	57.91% (-0.69)	58.44% (-0.16)
CIFAR-100	CNN	18.80%	15.58% (-3.22)	14.04% (-4.76)	18.33% (-0.47)	18.73% (-0.07)
	Res50	34.81%	33.17% (-1.64)	31.72% (-3.09)	34.40% (-0.41)	34.59% (-0.22)
Shakespeare	RNN	28.54%	26.50% (-2.04)	25.11% (-3.43)	28.09% (-0.45)	28.42% (-0.12)
Average		53.51%	50.89% (-2.62)	48.80% (-4.71)	53.04% (-0.45)	53.32% (-0.19)

TABLE VI
GLOBAL TEST ACCURACY WITH HIGHER PRIVACY BUDGETS FOR DIFFERENT TRAINING METHODS ON CNN

Privacy Budget	Method (DP-)	MNIST	CIFAR-10	CIFAR-100
0.1	FedAvg [24]	88.34%	34.49%	10.36%
	FedASGP	91.16%	40.96%	14.34%
0.2	FedAvg	91.68%	42.88%	16.17%
	FedASGP	93.24%	46.86%	19.44%
0.5	FedAvg	93.87%	48.36%	20.21%
	FedASGP	94.75%	51.37%	22.63%
1	FedAvg	95.12%	52.73%	22.74%
	FedASGP	95.83%	54.56%	25.04%
2	FedAvg	96.27%	55.87%	24.97%
	FedASGP	96.68%	57.12%	26.77%
4	FedAvg	96.89%	58.10%	26.53%
	FedASGP	97.32%	59.83%	27.87%

with $\varepsilon = \{0.1, 0.2, 0.5, 1, 2, 4\}$ in the above experimental environment. Note that a higher global test accuracy in the experimental results indicates better applicability.

As the privacy budget increases, DP-FedASGP consistently outperforms DP-FedAvg in global test accuracy. When $\varepsilon = 0.1$, DP-FedASGP achieves higher global test accuracy than DP-FedAvg by approximately 2.82%, 6.47%, and 3.98% on the MNIST and CIFAR-10/100 datasets, respectively. When $\varepsilon = 4$, DP-FedASGP outperforms DP-FedAvg by about 0.43%,

1.73%, and 1.34% on the three datasets. However, as ε increases from 0.1 to 4, the performance gap between DP-FedASGP and DP-FedAvg gradually diminishes. Although DP-FedASGP may result in a slight decrease in privacy protection, it simplifies the gradients of the model during training. When training with low privacy budgets, DP-FedASGP can provide sufficient privacy protection to the gradients. This makes it challenging for adversaries to infer sensitive information, even if some gradient information is exposed through membership inference attacks. Since DP-FedASGP only introduces noise perturbation into partial significant gradients in the current training round and

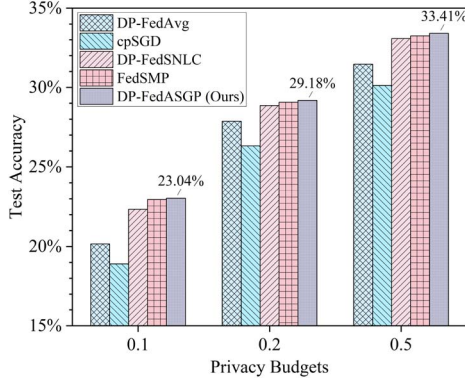


Fig. 7. Global test accuracy on Shakespeare dataset.

does not change the amount of noise perturbation in essence. DP-FedASGP can still perform initial perturbation for gradients according to the privacy budgets. Therefore, the lower the privacy budgets, the better the availability of DP-FedASGP. Additionally, DP-FedASGP can maintain excellent applicability even under higher privacy budgets.

E. Summary of Experiments

We implemented the experimental environments of DP-FL, in which we conducted comprehensive experiments for evaluating the performance of DP-FedASGP by comparing with baselines such as DP-FedAvg, cpSGD, DP-FedSNLC, and FedSMP. These experiments were carried out on the MNIST, CIFAR-10/100, and Shakespeare datasets. The experimental results show that the average global test accuracy of DP-FedASGP on the four datasets and three models is about 2.62%, 4.71%, 0.45%, and 0.19% higher than DP-FedAvg, cpSGD, DP-FedSNLC, and FedSMP, respectively. When training with low privacy budgets, DP-FedASGP can improve model accuracy while ensuring privacy protection, exploring a better balance between these two aspects efficiently. Even under higher privacy budgets, DP-FedASGP can still maintain excellent applicability. These improvements in DP-FedASGP can enhance both the privacy protection and model accuracy of the global model during the model training.

VI. CONCLUSION

Local gradients of DP-FL become excessively sparse in certain training rounds. Especially when training with low privacy budgets, there is a risk of introducing excessive noise into the uploaded gradients. This issue leads to a significant degradation in the accuracy of the global model. To effectively balance the privacy protection and model accuracy of DP-FL, we propose an approach called DP-FedASGP, which combines the idea of gradient sparsification and DP to achieve both gradient perturbation and gradient aggregation in DP-FL. Particularly, DP-FedASGP constructs a gradient perturbation method based on sparse vectors to evaluate and protect significant gradients in each training round. Subsequently, to dynamically calculate the aggregation weights of the gradients, DP-FedASGP employs

a dynamic aggregation weights calculation method based on the local loss function and the local data size. DP-FedASGP then formulates the global model gradient aggregation method to accelerate the convergence of the global model. Experiments on four datasets and three models manifest that DP-FedASGP can more effectively perturb significant gradients during each training round. Thus, DP-FedASGP can enhance the accuracy and availability of model training while ensuring privacy protection. Therefore, DP-FedASGP can effectively explore a better balance between privacy protection and model accuracy of DP-FL.

APPENDIX A PROOF OF THE THEOREM 1

According to Assumption 1, after introducing Laplace noise, the output function is

$$F'_{model}(\mathbf{D}) = F_{model}(\mathbf{D}) + \left(\text{Laplace}_1 \left(\frac{\Delta S}{\varepsilon} \right), \text{Laplace}_2 \left(\frac{\Delta S}{\varepsilon} \right), \dots, \text{Laplace}_d \left(\frac{\Delta S}{\varepsilon} \right) \right), \quad (21)$$

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

$$\Delta S = \max_{\mathbf{D}, \mathbf{D}'} \left(\sum_{i=1}^d |\Delta x_i| \right). \quad (22)$$

Because the output function $F'_{model}(\mathbf{D})$ satisfies the definition of ε -DP, which is given by (4). Then we can get

$$\Pr [F'_{model}(\mathbf{D}) = O] \leq e^\varepsilon \Pr [F'_{model}(\mathbf{D}') = O]. \quad (23)$$

To prove Theorem 1, we need to prove the validity of (23). As we aggregate the global gradient based on the gradient weights, then we have

$$F_{model}(\mathbf{D}') = (x'_1, x'_2, \dots, x'_d)^T = (x_1 + \Delta x_1, x_2 + \Delta x_2, \dots, x_d + \Delta x_d)^T, \quad (24)$$

according to (24) we can get

$$\Delta S = \max_{\mathbf{D}, \mathbf{D}'} \left(\sum_{i=1}^d |x_i - x'_i| \right). \quad (25)$$

Since DP-FedASGP needs to introduce Laplacian noise into partial gradients, datasets \mathbf{D} and \mathbf{D}' need to satisfy the selection of the partial gradients. Therefore, we define $\omega \in [0, 1]$ as the gradient selection coefficient. As $\omega \rightarrow 1$, more gradients are selected. Thus, for any domain function with input datasets \mathbf{D} and \mathbf{D}' , we have

$$F_{model}(\mathbf{D}) = (x_1, x_2, \dots, x_{\omega d}, \dots, x_d)^T, \quad (26)$$

$$F_{model}(\mathbf{D}') = (x'_1, \dots, x'_{\omega d}, \dots, x'_d)^T \\ = (x_1 + \Delta x_1, \dots, x_{\omega d} + \Delta x_{\omega d}, \dots, x_d)^T, \quad (27)$$

with input datasets \mathbf{D} , \mathbf{D}' and sensitivity ΔS , we can get

$$\Delta S_N = \max_{\mathbf{D}, \mathbf{D}'} \left(\sum_{i=1}^{\omega d} |x_i - x'_i| \right) \\ = \max_{\mathbf{D}, \mathbf{D}'} \left(\sum_{i=1}^{\omega d} |\Delta x_i| \right) \leq \Delta S. \quad (28)$$

According to Assumption 2, we can get $F_{model}(\mathbf{D}) = (0, 0, \dots, 0)^T$, $F_{model}(\mathbf{D}') = (\Delta x_1, \Delta x_2, \dots, \Delta x_{\omega d}, \dots, 0)^T$. When $O = (y_1, y_2, \dots, y_d)^T$, we have

$$\Pr[F'_{model}(\mathbf{D}) = O] = \prod_{i=1}^{\omega d} \frac{\varepsilon}{2\Delta S_N} e^{-\frac{\varepsilon}{2\Delta S_N} |\gamma_i|}, \quad (29)$$

$$\Pr[F'_{model}(\mathbf{D}') = O] = \prod_{i=1}^{\omega d} \frac{\varepsilon}{2\Delta S_N} e^{-\frac{\varepsilon}{2\Delta S_N} |\Delta x_i - y_i|}. \quad (30)$$

Then we can get

$$\frac{\Pr[F'_{model}(\mathbf{D}) = O]}{\Pr[F'_{model}(\mathbf{D}') = O]} = \frac{\prod_{i=1}^{\omega d} \frac{\varepsilon}{2\Delta S_N} e^{-\frac{\varepsilon}{2\Delta S_N} |y_i|}}{\prod_{i=1}^{\omega d} \frac{\varepsilon}{2\Delta S_N} e^{-\frac{\varepsilon}{2\Delta S_N} |\Delta x_i - y_i|}} \\ = e^{\frac{\varepsilon}{2\Delta S_N} \sum_{i=1}^{\omega d} (|\Delta x_i - y_i| - |y_i|)}. \quad (31)$$

To prove the validity of (23), we need to prove $\sum_{i=1}^{\omega d} (|\Delta x_i - y_i| - |y_i|) \leq \Delta S_N$. For each $|\Delta x_i - y_i| - |y_i|$, according to the absolute inequality, we have

$$\sum_{i=1}^{\omega d} (-|\Delta x_i|) \leq \sum_{i=1}^{\omega d} (|\Delta x_i - y_i| - |y_i|) \leq \sum_{i=1}^{\omega d} (|\Delta x_i|), \quad (32)$$

and

$$\sum_{i=1}^{\omega d} (|\Delta x_i|) \leq \max_{\mathbf{D}, \mathbf{D}'} \left(\sum_{i=1}^{\omega d} |\Delta x_i| \right) = \Delta S_N \leq \Delta S, \quad (33)$$

according to (29), (30) and (33), we can get

$$\sum_{i=1}^{\omega d} (|\Delta x_i - y_i| - |y_i|) \leq \Delta S_N \leq \Delta S. \quad (34)$$

We can get $\sum_{i=1}^{\omega d} (|\Delta x_i - y_i| - |y_i|) \leq \Delta S_N$ from (34). Therefore, we can prove that (10) exit and $F'_{model}(\mathbf{D})$ can satisfy the definition of DP, i.e. $\Pr[F'_{model}(\mathbf{D}) = O] \leq e^\varepsilon \Pr[F'_{model}(\mathbf{D}') = O]$ holds. Thus, **Theorem 1** concludes.

APPENDIX B

PROOF OF THE THEOREM 2

We prove Theorem 2 from both $\forall i R_i(\mathbf{D}) \geq R_i(\mathbf{D}')$ and $\forall i R_i(\mathbf{D}) \leq R_i(\mathbf{D}')$. If both $R_i(\mathbf{D}) \geq R_i(\mathbf{D}')$ and $R_i(\mathbf{D}) \leq R_i(\mathbf{D}')$ can prove Theorem 2, then Theorem 2 is true.

First, we assume that $\forall i R_i(\mathbf{D}) \geq R_i(\mathbf{D}')$. Then, (35) and (36) exist.

$$f_i(\mathbf{D}, \kappa) = \Pr[R_i(\mathbf{D}) + \alpha < \lambda + \kappa], \quad (35)$$

$$g_i(\mathbf{D}, \kappa) = \Pr[R_i(\mathbf{D}) + \alpha \geq \lambda + \kappa], \quad (36)$$

where κ is the parameter input for the function $f_i(\mathbf{D}, \kappa)$. Then we have

$$f_i(\mathbf{D}, \kappa) = \Pr[R_i(\mathbf{D}) + \alpha < \lambda + \kappa] \\ \leq \Pr[R_i(\mathbf{D}') + \alpha < \lambda + \kappa] \\ = f_i(\mathbf{D}', \kappa), \quad (37)$$

$$g_i(\mathbf{D}, \kappa) = \Pr[R_i(\mathbf{D}) + \alpha \geq \lambda + \kappa] \\ \leq \Pr[R_i(\mathbf{D}') + \alpha + \chi \geq \lambda + \kappa] \\ \leq e^{\varepsilon_1/q} \Pr[R_i(\mathbf{D}') + \alpha \geq \lambda + \kappa] \\ = e^{\varepsilon_1/q} g_i(\mathbf{D}', \kappa), \quad (38)$$

according to (37) and (38), we can get

$$\Pr[\mathcal{M}(\mathbf{D})] \\ \leq \int_{-\infty}^{+\infty} \Pr[\kappa = \beta] \prod_{j \in i} f_j(\mathbf{D}', \kappa) \prod_{j \notin i} e^{\varepsilon_1/q} g_j(\mathbf{D}', \kappa) d\kappa \\ \leq (e^{\varepsilon_1/q})^q \Pr[\mathcal{M}(\mathbf{D}')] \leq e^{\varepsilon_1 + \varepsilon_2} \Pr[\mathcal{M}(\mathbf{D}')]. \quad (39)$$

Following these steps, (39) can satisfy Definition 4. Therefore, $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$ holds.

Next, we assume that $\forall i R_i(\mathbf{D}) \leq R_i(\mathbf{D}')$. Then, we have $\forall i R_i(\mathbf{D}) \geq R_i(\mathbf{D}') - \chi$. Following the steps above, we have

$$f_i(\mathbf{D}, \kappa - \chi) = \Pr[R_i(\mathbf{D}) + \alpha < \lambda + \kappa - \chi] \\ \leq \Pr[R_i(\mathbf{D}') - \chi + \alpha < \lambda + \kappa - \chi] \\ = f_i(\mathbf{D}', \kappa), \quad (40)$$

$$g_i(\mathbf{D}, \kappa - \chi) = \Pr[R_i(\mathbf{D}) + \alpha \geq \lambda + \kappa - \chi] \\ \leq \Pr[R_i(\mathbf{D}') + \alpha \geq \lambda + \kappa - \chi] \\ \leq e^{\varepsilon_1/q} \Pr[R_i(\mathbf{D}') + \alpha \geq \lambda + \kappa] \\ = e^{\varepsilon_1/q} g_i(\mathbf{D}', \kappa), \quad (41)$$

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

$$\Pr[\mathcal{M}(\mathbf{D})] = \int_{-\infty}^{+\infty} \Pr[\kappa = \beta + \chi] \prod_{j \in i} f_j(\mathbf{D}', \kappa - \chi) \\ \times \prod_{j \notin i} e^{\varepsilon_1/q} g_j(\mathbf{D}', \kappa - \chi) d\kappa \\ \leq \int_{-\infty}^{+\infty} e^{\varepsilon_2} \Pr[\kappa = \beta] \prod_{j \in i} f_j(\mathbf{D}', \kappa) \\ \times \prod_{j \notin i} e^{\varepsilon_1/q} g_j(\mathbf{D}', \kappa) d\kappa \\ \leq (e^{\varepsilon_1/q})^q e^{\varepsilon_2} \Pr[\mathcal{M}(\mathbf{D}')] = e^{\varepsilon_1 + \varepsilon_2} \Pr[\mathcal{M}(\mathbf{D}')]. \quad (42)$$

Following the above steps, (42) can satisfy Definition 4, $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$ still holds. From both $\forall i R_i(\mathbf{D}) \geq R_i(\mathbf{D}')$ and $\forall i R_i(\mathbf{D}) \leq R_i(\mathbf{D}')$, we can always get the total privacy budget $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$. In this way, **Theorem 2** concludes.

APPENDIX C

PROOF OF THE THEOREM 3

We prove that the global model parameters W_t in T training rounds can form a Cauchy sequence, thereby further demonstrating the convergence of the DP-FedASGP. The Cauchy sequence is a special case of a real number sequence that plays an important role in mathematical analysis. A sequence of real numbers x_n is called a Cauchy sequence if, for any given positive real number $\rho > 0$, there exists a positive integer ψ such that for all $m, n > \psi$, the distance between any two terms in the sequence $|x_n - x_m|$ is less than ρ .

In other words, for any given precision requirement ε , when the number of terms in the sequence is sufficiently large, the distance between any two terms in the sequence is close enough. This means that as the number of terms in the sequence increases, the differences between the terms become smaller and smaller, eventually approaching a limit. According to Assumption 4, (16), and (19) we can get

$$\begin{aligned} \|g_t\| &= \left\| \frac{1}{k} \sum_{i=1}^k \nabla L_i(W_t) \gamma_t(i) \right\| \\ &\leq \frac{1}{k} \sum_{i=1}^k \|\nabla L_i(W_t) \gamma_t(i)\| \leq \frac{1}{k} \sum_{i=1}^k M_i. \end{aligned} \quad (43)$$

Let $M = \max\{M_1, M_2, \dots, M_k\}$ represent the maximum norm of all client local gradients. Therefore, we have

$$\|g_t\| \leq \frac{1}{k} \cdot k \cdot M = M. \quad (44)$$

Thus, the global gradient g_t is also bounded. For all t , there is a constant M that makes $\|g_t\| \leq M$. This means that we can set the gradient clipping threshold C to a constant M .

Now that we have proved that the global gradient g_t is bounded. Next, we will prove that W_t is a Cauchy sequence, i.e., for any given $\rho > 0$, there exists a positive integer ψ . For any $m, n > \psi$, we have $\|W_m - W_n\| < \rho$. According to the bounded properties of the gradient and the conditions of the noise term, we have

$$\begin{aligned} \|W_m - W_n\| &= \|(W_m - W_{m-1}) + (W_{m-1} - W_{m-2}) \cdots + (W_{n+1} - W_n)\| \\ &\leq \eta(\|g_m\| + N) + \eta(\|g_{m-1}\| + N) \cdots + \eta(\|g_n\| + N) \\ &\leq \eta(M + N)(m - n), \end{aligned} \quad (45)$$

where N is noise. We can choose a small enough learning rate η such that $\eta(M + N) < \rho$. Suppose we choose $\eta = \frac{\rho}{2(M+N)(m-n)}$. Then we have

$$\|W_m - W_n\| < \eta(M + N)(m - n) = \frac{\rho}{2}. \quad (46)$$

Therefore, for a sufficiently large ψ , we have $\|W_m - W_n\| < \rho$, which proves that W_t is a Cauchy sequence, converging to a finite value. In this way, **Theorem 3** concludes.

REFERENCES

- [1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 12:1–12:19, Mar. 2019.
- [2] H. Hu, Z. Salicic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Comput. Surv.*, vol. 54, no. 11s, pp. 235:1–235:37, 2022.
- [3] O. Shahid, S. Pouriyeh, R. M. Parizi, Q. Z. Sheng, G. Srivastava, and L. Zhao, "Communication efficiency in federated learning: Achievements and challenges," Jul. 2021, *arXiv:2107.10996*.
- [4] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017.
- [5] J. Jordon and J. Yoon, "PATE-GAN: Generating synthetic data with differential privacy guarantees" in *Proc. 7th Int. Conf. Learn. Representations*, 2019.
- [6] 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," in *Proc. 10th Int. Conf. Learn. Representations (ICLR)*, 2022.
- [7] S. Weng et al., "Privacy-preserving federated learning based on differential privacy and momentum gradient descent," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2022, pp. 1–6.
- [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 *Proc. 21st Int. Symp. Theory, Algorithmic Found., Protocol Des. Mobile Netw. Mobile Comput.*, Virtual Event, USA, New York, NY, USA: ACM, 2020, pp. 261–270.
- [9] L. Cui, X. Su, Y. Zhou and Y. Pan, "Slashing communication traffic in federated learning by transmitting clustered model updates," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2572–2589, Aug. 2021.
- [10] J. Ding, J. Wang, G. Liang, J. Bi, and M. Pan, "Towards plausible differentially private admm based distributed machine learning," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Virtual Event Ireland, New York, NY, USA: ACM, 2020.
- [11] Y. Mao et al., "SAFARI: Sparsity-enabled federated learning with limited and unreliable communications," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4819–4831, May 2024.
- [12] S. Bibikar, H. Vikalo, Z. Wang, and X. Chen, "Federated dynamic sparse training: Computing less, communicating less, yet learning better," *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 6, 2022, Art. no. 6.
- [13] R. Dai, L. Shen, F. He, X. Tian, and D. Tao, "DisPFL: Towards communication-efficient personalized federated learning via decentralized sparse training," in *Proc. 39th Int. Conf. Mach. Learn. (PMLR)*, Jun. 2022, pp. 4587–4604.
- [14] L. Lyu, "DP-SIGNSGD: When efficiency meets privacy and robustness," May 2021, *arXiv:2105.04808*.
- [15] B. Zhao, K. Fan, K. Yang, Z. Wang, H. Li, and Y. Yang, "Anonymous and privacy-preserving federated learning with industrial big data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6314–6323, Sep. 2021.
- [16] J. Wang, S. Guo, X. Xie, and H. Qi, "Protect privacy from gradient leakage attack in federated learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2022, pp. 580–589.
- [17] M. Asad, A. Moustafa, and M. Aslam, "CEEP-FL: A comprehensive approach for communication efficiency and enhanced privacy in federated learning," *Appl. Soft Comput.*, vol. 104, p. 107235, Jun. 2021.
- [18] Y. Li, Y. Feng, and Q. Qian, "FDPBoost: Federated differential privacy gradient boosting decision trees," *J. Inf. Secur. Appl.*, vol. 74, p. 103468, May 2023.
- [19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, PMLR, 2017, pp. 1273–1282.
- [20] J. Neri, P. Depalle and R. Badeau, "Approximate inference and learning of state space models with Laplace noise," *IEEE Trans. Signal Process.*, vol. 69, pp. 3176–3189, 2021.
- [21] Z. Chuanxin, S. Yi, and W. Degang, "Federated learning with Gaussian differential privacy," in *Proc. 2nd Int. Conf. Robot., Intell. Control Artif. Intell.*, Shanghai China, New York, NY, USA: ACM, 2020, pp. 296–301.
- [22] 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. Hoboken, NJ, USA: Wiley, 2023, pp. 169–213.

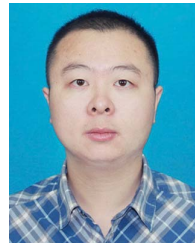
- [23] Q. Yang, X. Du, A. Liu, N. Wang, W. Wang, and X. Wu, "AdaSTopk: Adaptive federated shuffle model based on differential privacy," *Inf. Sci.*, vol. 642, p. 119186, Sep. 2023.
- [24] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Vienna, Austria, New York, NY, USA: ACM, 2016, pp. 308–318.
- [25] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpSGD: Communication-efficient and differentially-private distributed SGD," in *Proc. Neural Inf. Process. Syst.*, Montreal, Canada: Curran Associates, Inc., 2018, pp. 7575–7586.
- [26] Y. Gu, Y. Bai, and S. Xu, "CS-MIA: Membership inference attack based on prediction confidence series in federated learning," *J. Inf. Secur. Appl.*, vol. 67, p. 103201, Jun. 2022.
- [27] 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," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA: Internet Society, 2019, doi: 10.14722/ndss.2019.23119.
- [28] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Secur. Priv. (SP)*, San Francisco, CA, USA, 2019, pp. 739–753.
- [29] T. Hoang, A. A. Yavuz, and J. G. Merchan, "A secure searchable encryption framework for privacy-critical cloud storage services," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 1675–1689, Nov./Dec. 2021.
- [30] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Trans. Parallel Distrib.*, vol. 27, no. 2, pp. 340–352, Feb. 2016.
- [31] J. Hu et al., "Shield against gradient leakage attacks: Adaptive privacy-preserving federated learning," in *IEEE/ACM Trans. Netw.*, vol. 32, no. 2, pp. 1407–1422, Apr. 2024.
- [32] L. Song, R. Shokri, and P. Mittal, "Privacy risks of securing machine learning models against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 241–257.
- [33] S. Reddi et al., "Adaptive federated optimization," in *Proc. 9th Int. Conf. Learn. Representations (ICLR)*, 2021, doi: 10.48550/arXiv.2003.00295.
- [34] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010, *arXiv:1812.01097*.
- [35] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.
- [36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
- [37] C. Dwork et al., "The algorithmic foundations of differential privacy," *Found. Trends® Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.
- [38] R. Hu, Y. Guo and Y. Gong, "Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8242–8255, Aug. 2024.
- [39] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," Sep. 2019.



Benteng Zhang (Student Member, IEEE) received the B.S. degree in software engineering from the College of Computer Science and Technology, Qingdao University, Qingdao, China, in 2021. He is currently working toward the Ph.D. degree with the College of Computer Science and Software Engineering, Hohai University, Nanjing. His research interests include distributed machine learning, edge computing, and federated learning.



system. He is a Senior Member of China Computer Federation and Chinese Association of Automation.



Xiaoming He (Member, IEEE) received the Ph.D. degree in computer science and software engineering from Hohai University, Nanjing, China, in 2023. He is currently a Lecturer with the College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China. Prior to work, he was a Visiting Research Fellow with Singapore University of Technology and Design. His research interests include edge intelligence and FPGA-based AI accelerator.



THE COMPUTER SOCIETY.

Huawei Huang (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of Aizu, Japan, in 2016. He is currently an Associate Professor with Sun Yat-Sen University. His research interests include federated learning, blockchain, and distributed systems. He received the Best Paper Awards from TrustCom2016 and IEEE OJ-CS. He has served as a Lead Guest Editor for multiple special issues organized at IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and IEEE OPEN JOURNAL OF



network trust and security, distributed algorithms, and cloud computing. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award. He was an IEEE Computer Society Distinguished Visitor, an ACM Distinguished Speaker, and the Chair for the IEEE Technical Committee on Distributed Processing. He is a fellow of the AAAS.

Jie Wu (Fellow, IEEE) received the Ph.D. degree in computer engineering from Florida Atlantic University, Boca Raton, FL, USA, in 1989. He is the Director of the Center for Networked Computing and a Laura H. Carnell Professor with the Temple University, Philadelphia, PA, USA, and also serves as the Director of International Affairs, College of Science and Technology. He regularly publishes in scholarly journals, conference proceedings, and books. His research interests include mobile computing and wireless networks, routing protocols,