Exploring the Privacy-Accuracy Trade-off Using Adaptive Gradient Clipping in Federated Learning

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

***Abstract*—In Differentially Private Federated Learning (DP-FL), gradient clipping can prevent excessive noise from being added to the gradient and ensure that the impact of noise is within a controllable range. However, state-of-the-art methods adopt fixed or imprecise clipping thresholds for gradient clipping, which is not adaptive to the changes in the gradients. This issue can lead to a significant degradation in accuracy while training the global model. To this end, we propose Differential Privacy Federated Adaptive gradient Clipping based on gradient Norm (DP-FedACN). DP-FedACN can calculate the decay rate of the clipping threshold by considering the overall changing trend of the gradient norm. Furthermore, DP-FedACN can accurately adjust the clipping threshold for each training round according to the actual changes in gradient norm, clipping loss, and decay rate. Experimental results demonstrate that DP-FedACN can maintain privacy protection performance similar to that of DP-FedAvg under member inference attacks and model inversion attacks. DP-FedACN significantly outperforms DP-FedAGNC and DP-FedDDC in privacy protection metrics. Additionally, the test accuracy of DP-FedACN is approximately 2.61%, 1.01%, and 1.03% higher than the other three baseline methods, respectively. DP-FedACN can improve the global model training accuracy while ensuring the privacy protection of the model. All experimental results demonstrate that the proposed DP-FedACN can help find a fine-grained privacy-accuracy trade-off in DP-FL.**

***Index Terms*—Federated learning, differential privacy, gradient clipping**

I. Introduction

Federated Learning (FL) enables numerous edge devices to collaboratively train a global model without sharing private data [1]. However, as illustrated in *Challenge 1* in Fig. 1, state-of-the-art studies indicate that adversaries can still infer sensitive user data characteristics by Membership Inference Attacks (MIAs). Therefore, some existing methods add Differential Privacy (DP) noise to the gradients to enhance privacy protection [2, 36]. Differentially Private Federated Learning (DP-FL) requires clipping gradients before adding noise to control the amount of noise added to the gradients [3, 4]. As depicted in *Challenge 2* in Fig. 1, most existing methods use fixed or imprecise clipping thresholds for gradient clipping. However, fixed or imprecise clipping thresholds are not adaptive to the changes in the gradients [5]. A too-small clipping threshold discards important information in the gradients, while a too-large clipping threshold adds excessive noise to the gradients. Both cases ultimately lead to a significant degradation in the accuracy and availability of the global model. Therefore, it is crucial to find an appropriate clipping threshold to better solve the privacy-accuracy trade-off.

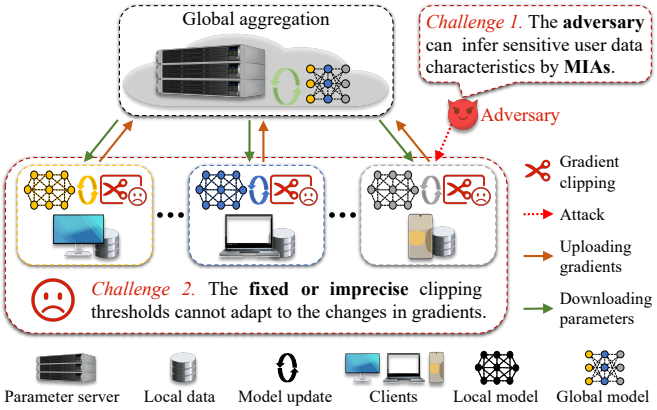


Fig. 1. Two challenges in handling the gradient clipping for Federated Learning.

State-of-the-art works have proposed using dynamic clipping thresholds instead of fixed ones, providing new insights for our study. For example, DP-FedAGNC utilizes the average ℓ_2-norm of gradients from the previous training round as the gradient clipping threshold for the next training round [6]. However, the average ℓ_2-norm of gradients cannot accurately reflect the changes in gradients uploaded by each client, and the calculation of the average requires an additional privacy budget [7]. This may negatively impact the accuracy of the global model. Besides, DP-FedAMVC employs a linear decay function to calculate the gradient clipping threshold for the t-th training round [8]. The decay function operates independently of the training data and does not require an additional privacy budget. However, we found that the number of FL training rounds in DP-FedAMVC significantly influences the

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 Provence under Grant BA2021002. *(Corresponding author: Yingchi Mao.)*

Yingchi Mao, Ping Ping, and Benteng Zhang are with the College of Computer Science and Software Engineering, Hohai University, Nanjing 211100, China (e-mail: yingchimao@hhu.edu.cn; pingpingnjust@163.com; 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, 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).

model accuracy in practical scenarios. Moreover, DP-FedDDC uses a near-linear decay function to calculate the gradient clipping threshold [18]. However, DP-FedDDC shows good performance only when the privacy budget is small. These methods employ the idea of dynamically adjusting clipping thresholds for each training round. However, although these methods consider changes in gradients, the clipping thresholds calculated in each training round are not precise enough and cannot accurately adapt to the changes in gradients.

Motivated by the works mentioned, we aim to explore a better privacy-accuracy trade-off by adjusting clipping thresholds in an adaptive manner. To this end, we propose an approach, known as Differential Privacy Federated Adaptive gradient Clipping based on gradient Norm (DP-FedACN), which can address the problem of fixed or imprecise clipping thresholds that are not adaptive to changes in gradients. In each training round, DP-FedACN calculates the decay rate of the clipping threshold by exploiting the overall trend of gradient norm changes. Subsequently, DP-FedACN can construct an adaptive clipping threshold computation method by utilizing the actual changes in gradient norm, clipping loss, and decay rate. In brief, by taking the trade-off between privacy protection and model accuracy into account, DP-FedACN can adaptively adjust the clipping thresholds for each training round.

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

- **Originality.** We prove that the gradient norm gradually decreases when training models using DP-SGD [9]. We build a clipping threshold-control mechanism for global decay, aiming to prevent sudden changes in the overall trend of the clipping threshold.
- **Methodology.** To accurately adjust the clipping thresholds, we construct an adaptive clipping threshold computation mechanism by considering the actual changes in gradient norm, clipping loss, and decay rate.
- **Effectiveness.** The experimental results indicate that DP-FedACN improves test accuracy by approximately 2.61%, 1.01%, and 1.03% compared to DP-FedAvg, DP-FedAGNC, and DP-FedDDC, respectively, while ensuring model privacy protection.

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-FedACN are discussed in Section IV. The experiments and analysis are given in Section V. Finally, we conclude with Section VI.

II. Related Work

In practical application environments, the data distribution across different clients varies significantly (i.e., non-IID), and FL models often experience constantly changing gradient norms during client updates due to heterogeneous data. This can lead to slow global model convergence and a significant degradation in model accuracy. From the perspective of "rectifying" data heterogeneity, Virtual Homogeneity Learning (VHL) uses a virtual homogeneous dataset that contains no private information and is separable for FL [31]. This virtual dataset can be generated from pure noise shared across clients, aiming to calibrate the features from heterogeneous clients. To mitigate the impact of heterogeneous data on model accuracy by generating improved local models, Tang *et al.* proposed FedImpro [30], which decouples the model into high-level and low-level components and trains the high-level part on reconstructed feature distributions. FedImpro can enhance generalization contribution and reduce gradient dissimilarity in FL. Additionally, GossipFL constructs a communication-efficient decentralized FL framework using a sparsification algorithm [32]. Thus, GossipFL can counter the changing gradient norm during client updates by accelerating global model convergence and efficient communication. Furthermore, to enhance the privacy protection level in FL, Li *et al.* proposed Blockchain-Assisted Decentralized Federated Learning (BLADE-FL) [33]. In each training round, BLADE-FL broadcasts each client's trained model to other clients, aggregates the received models with the client's own model, and competes to generate a block before the next training round. At the same time, BLADE-FL alleviates training defects caused by lazy clients who steal other clients' trained models.

In DP-FL, the trade-off between privacy protection and model accuracy is a perpetual and hot topic [11–13]. Gradient clipping can directly prevent excessive noise from being added to the gradient to improve the global model training accuracy, which makes it one of the most effective ways to balance privacy and accuracy. Abadi *et al.* proposed DP-FedAvg (a DP-protected SGD algorithm) [10], which uses a fixed gradient clipping to clip gradients. Thus, DP-FedAvg can protect sensitive private data and prevent excessive noise from being added. However, using a fixed clipping threshold in DP-FedAvg can lead to the loss of important information in the gradient. This can lead to a degradation in the accuracy of the global model. Moreover, DP-FedGGC divides the original gradient into k groups and then calculates the ℓ_2-norm for each group [14]. Gradient clipping is performed for each of the k groups using k clipping thresholds. As a result, DP-FedGGC can achieve better performance than methods that use a fixed clipping threshold. However, DP-FedGGC requires manual grouping of the gradient information, so the grouping result is still discrete and is essentially not significantly different from the fixed clipping method. This leads to poor accuracy of the global model. Therefore, to the best of our knowledge, fixed clipping thresholds cannot adapt to changes in the gradient. Dynamically adjusting the clipping threshold to set an appropriate and accurate threshold for each training round is still a major challenge in DP-FL.

Recently, some studies have proposed using dynamic clipping thresholds instead of fixed clipping thresholds [6–8], providing a new perspective for our research. To save the cost of the privacy budget, Andrew *et al.* proposed gradient clipping based on specific quantiles of the gradient update norm distribution [37]. However, this method only considers one case (i.e., gradient norms less than or equal to the clipping threshold) and requires the addition of privacy noise twice on the client side. This can lead to imprecise dynamic updates of the clipping threshold and significantly increase the local computational overhead on the client side. Guo *et al.* confirmed that the main reason for the poor performance of gradient clipping is the use of a fixed threshold during training

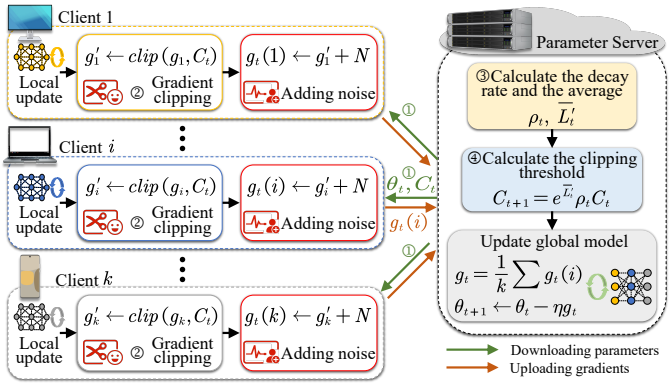


Fig. 2. The overview of proposed DP-FedACN.

[15]. This led them to overlook the dynamic nature of benign local updates during the convergence of the global model. The team proposed a method for dynamically adjusting the clipping threshold and demonstrated its convergence in a non-convex environment. Wang *et al.* proposed DP-FedMeta, a dynamic gradient clipping method [16]. DP-FedMeta can dynamically adjust the clipping thresholds by considering the ratio of the noise level to the gradient norm to balance model utility and privacy in federated meta-learning. However, although the above methods use the idea of dynamically adjusting the clipping threshold, they cannot accurately reflect the effect of gradient clipping. This leads to imprecise adjustments of the clipping threshold and sub-optimal model accuracy. This is because the gradient norm is different for different tasks and training processes. Therefore, finding a precise clipping threshold to balance privacy protection and model accuracy still remains a significant challenge in DP-FL.

III. Proposed Framework

A. System Model

As shown in Fig. 2, the system model includes a parameter server and a set of clients $\mathcal{K} = \{1, 2, \ldots, i, \ldots, k\}$ that collaboratively train a global model M with parameter θ. Client i uses its private data D_i to locally iterate E times to update its local model M_i. After local training, client i clips the gradient by exploiting the clipping threshold C_t. Then, the clipped gradients are perturbed with Gaussian noise and uploaded to the parameter server [17]. The parameter server aggregates the global gradient and updates the global model M. The main symbolic parameters are shown in Table I.

Specifically, in the t-th training round, the parameter server first sends the clipping threshold C_t and the global model parameter θ_t to each client. Next, client i updates the local model M_i and clips the gradient g by using C_t. After adding noise $N(0, \sigma^2)$ to the clipped gradient $g_t'(i)$, client i uploads the processed gradient $g_t(i)$ to the parameter server. After all clients upload the gradients, the parameter server calculates the clipping threshold decay rate ρ_t. Then, the parameter server calculates the average value $\overline{L_t'}$ of the derivative of the clipping loss function for k clients by exploiting the gradient norm. Thus, the parameter server calculates the gradient clipping threshold C_{t+1} for the $(t+1)$-th training round using ρ_t, C_t, and $\overline{L_t'}$. Meanwhile, the parameter server aggregates the global gradient g_t and updates the global model parameters θ_{t+1}. The model iterates through these steps for T rounds until the global model achieves the desired performance.

Before adding noise to the gradient, the original gradient g needs to be clipped using the clipping threshold C. The clipped gradient g' is given by

$$g' = \frac{g}{\max(1, \frac{\|g\|_2}{C})}. \tag{1}$$

B. Threat Model

Our threat model has similar assumptions to previous works [28, 29]. *1)* All clients participating in FL training are honest-but-curious. *2)* The parameter server is considered honest and trustworthy. *3)* External adversaries attempt to infer whether a given sample from an input dataset (i.e., the target dataset) belongs to the training dataset of the client model (i.e., the target model). Therefore, we use Membership Inference Attacks (MIAs) as the main attack method. Specifically, the adversaries access the client model by sending a series of queries and collecting the responses. Then, the adversaries apply analytical techniques and algorithms to infer whether certain membership or data features are present, aiming to determine if the query record belongs to the client model's training dataset. All devices participating in FL training (e.g., parameter servers and clients) complete the training process according to the FL protocol. Additionally, the communication channels between clients and the parameter server are not secure, and adversaries may intercept or disrupt data communication.

IV. Gradient Clipping and Global Decay Control

A. Clipping Threshold-Control Mechanism for Global Decay

In this section, we prove that the gradient norm gradually decreases with an increase in training rounds. The overall trend of the gradient norm should be roughly the same. To prevent sudden changes in the decay trend of the gradient norm due to malicious gradients, we use clipping threshold decay rate for global control. We have **Definition 1** and **Theorem 1**.

Definition 1. *Let the model $f : \mathbb{R} \mapsto \mathbb{R}$ denote a quadratic differentiable strictly convex function, then the second-order Taylor approximate expansion of the model f at x_{a-1} is defined as*

$$\begin{aligned} f(x) = f(x_{a-1}) &+ \nabla f(x_{a-1})^T (x - x_{a-1}) \\ &+ \frac{1}{2}(x - x_{a-1})^T H (x - x_{a-1}) \\ &+ o(\|x - x_{a-1}\|_2^2), \end{aligned} \tag{2}$$

where x_{a-1} is any point on the coordinate axis, H is the Hessian matrix, and $o(\|x - x_a\|_2^2)$ is a higher-order infinitesimal.

Theorem 1. *If client i trains the local model using the DP-SGD algorithm, then the gradient norm on a single client will gradually decrease with the increase in training rounds.*

Proof. As the model f is a strictly convex function, the Hessian matrix H is a symmetric and positive definite matrix. Furthermore, the DP-SGD algorithm updates the values at

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
θ	Global model parameters
η	Learning rate
ζ	Noise level
δ	Relaxation factor of noise
ρ_t	The adaptive decay rate of t-th training round
g_t	Global gradient in t-th training round
M	Global model
T	Training rounds
E	Local iterations
B	Local batch size
D_i	Local privacy dataset of client i
F_t	Decay coefficient of t-th training round
C_t	Clipped threshold of t-th training round
ΔS	Global sensitivity
$g_t(i)$	Upload gradient of client i
$L_t(i)$	Clipping loss of client i
D, D'	Sibling datasets
$N(\mu, \sigma^2)$	Gaussian noise

the point x_a by exploiting the information from the previous iteration round, which is given by

$$x_a = x_{a-1} - \eta \left(\nabla f\left(x_{a-1}\right) + N(\mu, \sigma^2) \right), \tag{3}$$

where η is the learning rate. Without loss of generality, we let $N(\mu, \sigma^2)$ represent the noise that follows a Gaussian distribution with a mean of μ and a variance of σ^2. The gradient at the point x_a is given by

$$\begin{aligned} \nabla f\left(x_a\right) &= \nabla f\left(x_{a-1}\right) - \eta H \left(\nabla f\left(x_{a-1}\right) + N(\mu, \sigma^2) \right) \\ &= (I - \eta H) \nabla f\left(x_{a-1}\right) - \eta H N(\mu, \sigma^2), \end{aligned} \tag{4}$$

where I is the identity matrix. By taking the ℓ_2-norm on both sides of the equation, we derive the following inequality

$$\begin{aligned} \left\|\nabla f\left(x_a\right)\right\|_2 \leq \left\|(I - \eta H)\right\|_2 \left\|\nabla f\left(x_{a-1}\right)\right\|_2 \\ - \left\|\eta H N(\mu, \sigma^2)\right\|_2. \end{aligned} \tag{5}$$

Let γ_{min} and γ_{max} represent the minimum and maximum eigenvalues of H, respectively. Then, when $\eta \leq 1/\gamma_{\max}$, $\left\|(I - \eta H)\right\|_2$ is equivalent to $1 - \eta\gamma_{min}$. Thus, we find

$$\begin{aligned} &\left\|\nabla f(x_a)\right\|_2 - \left\|\nabla f(x_{a-1})\right\|_2 \\ &\leq \eta(\left\|H N(\mu, \sigma^2)\right\|_2 - \gamma_{\min} \left\|\nabla f(x_{a-1})\right\|_2). \end{aligned} \tag{6}$$

If the inequality $\left\|H N(\mu, \sigma^2)\right\|_2 - \gamma_{min} \left\|\nabla f\left(x_{a-1}\right)\right\|_2 \leq 0$ holds, then **Theorem 1** holds. Let $\Psi = H^T H$. Then, according to the transformation of the tail bound estimate formula for multivariate Gaussian variables, for any $t > 0$, we can get

$$\begin{aligned} &Pr\left[\delta \|H N(1, \sigma^2)\|_2 \geq \sqrt{\mathrm{tr}(\Psi) + 2\sqrt{\mathrm{tr}(\Psi) t} + 2\|\Psi\|_2 t}\right] \\ &\leq e^{-t}, \end{aligned} \tag{7}$$

where $tr(\cdot)$ is the trace operation of a matrix and $\Pr[\cdot]$ is the probability. Since $\left\|H N(\mu, \sigma^2)\right\|_2$ is equivalent to $\delta \|H N(1, \sigma^2)\|_2$, when $\gamma_{min} \left\|\nabla f\left(x_{a-1}\right)\right\|_2 \geq tr(\Psi)$, inequality $\left\|H N(\mu, \sigma^2)\right\|_2 - \gamma_{min} \left\|\nabla f\left(x_{a-1}\right)\right\|_2 \leq 0$ always holds true. Therefore, we prove **Theorem 1**. □

According to **Theorem 1**, the gradient norm decreases during model training. The trend of clipping threshold changes needs to be globally controlled from the overall training perspective. Therefore, we build a clipping threshold-control method for global decay. This method can control the global decay trend of the clipping threshold. In the early stages of training, we use a larger clipping threshold to accelerate model convergence. In the later stages, we use a smaller clipping threshold to improve the robustness of model training. In DP, since noise is scaled by exploiting the clipping threshold, a smaller clipping threshold can prevent excessive noise from damaging the original gradient. This makes gradient clipping more effective in each training round. As power functions align with our training strategies, we choose the basic power function $F = x^a$ to construct the decay gradient threshold function. We have **Definition 2** and **Theorem 2**.

Definition 2. *According to the property of the power function $F = x^a$, the attenuation coefficient F_t of the t-th training round is defined as*

$$F_t = \frac{1}{\sqrt{t}}. \tag{8}$$

Theorem 2. *If the decay gradient threshold function is constructed using a power function $F = x^a$, then the clipping threshold decay rate ρ_t is a bounded constant and does not affect the convergence of the global model.*

Proof. By mathematical induction, we can obtain the decay rate ρ_t for the t-th training round, which is given by

$$\rho_t = \begin{cases} \sqrt{\left(1 - \frac{1}{t}\right)} & \text{if } \mathrm{t} > 1 \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

Since the gradient norm on the client side keeps changing, we cannot obtain historical information about the changes in the gradient norm at the beginning of the training. This makes it difficult for ρ_0 to accurately fit the changes in the gradient norm. Therefore, we set $\rho_0 = 1$ to simplify the algorithm and ensure its stable operation in the early stages of training. Due to the number of training rounds $t \geq 1$ and $t \in \mathbb{Z}^+$, there exist minimum and maximum values for the decay rate $(\rho_t)_{\min}$ and $(\rho_t)_{\max}$ are given by

$$\begin{aligned} (\rho_t)_{\min} &= \lim_{t \to 2} \rho_t = \sqrt{\tfrac{1}{2}}, \\ (\rho_t)_{\max} &= \lim_{t \to \infty} \rho_t = 1. \end{aligned} \tag{10}$$

As the number of training rounds increases, ρ_t tends towards the constant value of 1. Thus, ρ_t is a strictly bounded decay value. The decay rate does not affect the convergence of the original model and we prove **Theorem 2**. □

B. Adaptive Clipping Threshold Computation Mechanism

During the model training process, the gradients on each client are constantly changing. The adaptively adjusted clipping threshold follows the trend of gradient changes. This allows the clipping threshold to reflect the gradient variations throughout the training process. For clipping threshold C_t, the clipping loss function $L_t(i)$ of client i is given by

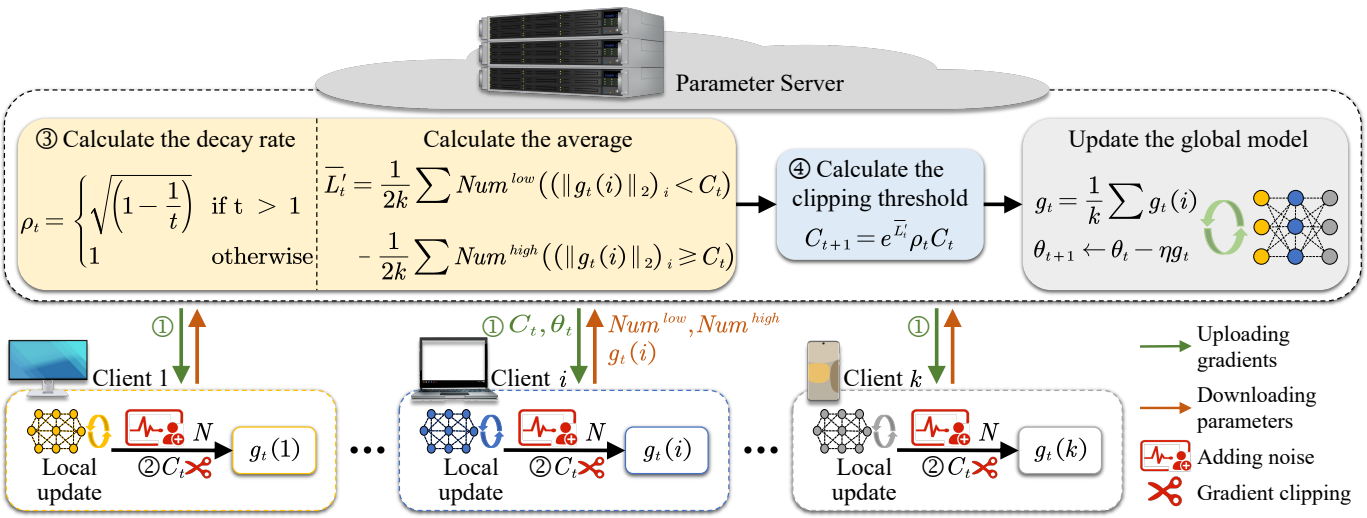


Fig. 3. Adaptive clipping threshold computation mechanism of DP-FedACN.

$$L_t(i) = \begin{cases} \frac{1}{2}(C_t - \|g_t(i)\|_2) & \text{if } \|g_t(i)\|_2 < C_t \\ \frac{1}{2}(\|g_t(i)\|_2 - C_t) & \text{otherwise.} \end{cases} \tag{11}$$

Now, to prove the effectiveness of the clipping loss function $L_t(i)$, we prove **Theorem 3**.

Theorem 3. *If $L_t(i)$ is used as the clipping loss function, then we can obtain an estimate of C_t using gradient descent.*

Proof. Let $L'_t(i)$ denote the derivative of $L_t(i)$. The expectation $\mathbb{E}_t(i)$ of $L'_t(i)$ is given by

$$\begin{aligned} \mathbb{E}_t(i) &= \frac{1}{2}Pr[\|g_t(i)\|_2 < C_t] - \frac{1}{2}Pr[C_t \leq \|g_t(i)\|_2] \\ &= Pr[\|g_t(i)\|_2 < C_t] - \frac{1}{2}, \end{aligned} \tag{12}$$

where $\mathbb{E}_t(i) \in [-0.5, 0.5]$. Since $L_t(i)$ is convex and the gradient is bounded by 1, we can obtain an estimated value of C_t with gradient descent. Thus, we prove **Theorem 3**. □

Since our system includes a parameter server and k clients. Let $\overline{L'_t}$ denote the average of L'_t, which is given by

$$\begin{aligned} \overline{L'_t} = \frac{1}{2k}(&\sum_{i=1}^{k} Num^{low}((\|g_t(i)\|_2)_i < C_t) \\ &- \sum_{i=1}^{k} Num^{high}((\|g_t(i)\|_2)_i \geqslant C_t)), \end{aligned} \tag{13}$$

where $Num(\phi)$ is a numerical value indicating whether the condition ϕ is satisfied, and it does not carry any information about the updated gradient. Therefore, uploading $Num(\phi)$ to the parameter server will not leak the client's privacy.

We aim to dynamically adjust the clipping threshold to follow the changes in the gradient during model training. Since $\overline{L'_t} \in [-0.5, 0.5]$, and the change in the clipping threshold may span multiple orders of magnitude, a linear function is not suitable for dynamically adjusting the clipping threshold. Therefore, we opt for an exponential function, which makes the adjustment of the clipping threshold more precise. The adaptively adjusted clipping threshold for the $(t+1)$-th training round is given by

$$C_{t+1} = e^{\overline{L'_t}}\rho_t C_t. \tag{14}$$

In the early stages of training, the change in the gradients uploaded by each client is usually significant. At this point, we need to protect the gradients rather than clip the gradients as much as possible. Only in this way can we retain more of the parameter update information carried by the gradients, thereby accelerating the convergence of the global model. Therefore, to avoid impacting the aggregation effectiveness of the global model in the first training round, we define the initial clipping threshold C_0 as the average gradient norm of the first training round, which is given by

$$C_0 = \frac{1}{k}\sum_{i=1}^{k}(\|g_t(i)\|_2)_i. \tag{15}$$

C. Algorithm Design

The gradient perturbation and aggregation process of DP-FedACN can be divided into two key stages.

1) Build A Clipping Threshold-Control Mechanism. In the t-th training round, the clipping threshold decay rate ρ_t is calculated by (9) on the parameter server.

2) Construct An Adaptive Clipping Threshold Computation Mechanism. In the t-th training round, each client in $\mathcal{K}$ uploads its gradient $g_t(i)$ to the parameter server. The parameter server can calculate the average value $\overline{L'_t}$ of $L'_t(i)$. With ρ_t, C_t, and $\overline{L'_t}$, the clipping threshold for the next round C_{t+1} is calculated by (14) on the parameter server.

In the t-th training round, the parameter server first sends C_t and θ_t to each client. Client i clips the gradient $g_t(i)$ by exploiting C_t and adds noise N into the clipped gradient $g_t(i)$ after updating the local model M_i (lines 13-28). Then, client i uploads its gradient $g_t(i)$ after adding noise. The parameter server calculates $L'_t(i)$, the average value $\overline{L'_t}$, and the decay

Algorithm 1: DP-FedACN

Input: k clients, σ, δ, η, T, D, M, B, E
Output: M

1 **begin**
2 Initialize M, M_i
3 **for** *each communication round* $t \in T$ **do**
4 **for** *each client* i *in parallel* **do**
5 $\theta_t(i) \leftarrow M_i$
6 $g_t(i), Num_i^{low}, Num_i^{high} \leftarrow clientTrain(\theta_t(i), D_i, C_t)$
7 **end**
8 $\overline{L_t'} \leftarrow \frac{1}{2k}(\sum Num_i^{low} - \sum Num_i^{high})$
9 $F_t \leftarrow \frac{1}{\sqrt{t}}, \rho_t \leftarrow \sqrt{\left(1 - \frac{1}{t}\right)}$
10 $C_{t+1} \leftarrow \boldsymbol{e}^{\overline{L_t'}} \rho_t C_t$
11 $g_t \leftarrow \frac{1}{k} \sum g_i$
12 $\theta_{t+1} \leftarrow \theta_t - \eta g$
13 **end**
14 **return** M
15 **end**
16 **function** $clientTrain(\theta, D, C)$
17 **begin**
18 **for** *each local epoch* $e \in E$ **do**
19 $g_e \leftarrow \nabla L(\theta)$
20 $\theta_i \leftarrow \theta_i - \eta g_e$
21 **end**
22 $g \leftarrow \sum g_e$
23 $g' \leftarrow \frac{g}{\max\left(1, \frac{\|g\|_2}{C}\right)}$
24 **if** $\|g\|_2 \geq C$ **then**
25 $Num^{high}\left(\|g\|_2 \geq C\right) \leftarrow 1$
26 **else**
27 $Num^{low}\left(\|g\|_2 < C\right) \leftarrow 1$
28 **end**
29 $g \leftarrow g' + N(0, \sigma^2)$
30 **return** $g, Num^{low}\left(\|g\|_2 < C\right), Num^{high}\left(\|g\|_2 \geq C\right)$
31 **end**

rate ρ_t (lines 7-8). With ρ_t, C_t, and $\overline{L_t'}$, the clipping threshold for the next round C_{t+1} is calculated (line 9). Finally, the parameter server aggregates the global gradient g_t and updates the global model parameters θ_{t+1} (lines 10-11). In the $(t+1)$-th training round, the parameter server sends C_{t+1} and θ_{t+1} to each client. The details are shown in ***Algorithm 1***.

D. Complexity Analysis

Algorithmic Complexity. In DP-FedACN, each client iterates E times to update the local model. The algorithmic complexity is $\mathcal{O}(\sum_{e=1}^{E} |\nabla|_e) = \mathcal{O}(E)$. Our system model consists of k clients and the algorithmic complexity for computing local gradients is $\mathcal{O}(k)$. Since all clients are training in parallel, the algorithmic complexity becomes $\mathcal{O}(1)$. We have T training rounds and the overall complexity of DP-FedACN is $\mathcal{O}(ET)$. Since $E \ll T$, the overall complexity of the DP-FedACN algorithm is $\mathcal{O}(n)$.

Gradient Clipping. According to ***Algorithm 1***, the time complexity of gradient clipping consists of three steps.

- *The time complexity of gradient computation.* During each local epoch, the client needs to compute the gradient g_e. Assuming the number of model parameters is P (i.e., the gradient vector has a dimension of P), the complexity of computing the gradient in each epoch is $\mathcal{O}(P)$. After E epochs, the cumulative complexity of the gradient g is $\mathcal{O}(E \cdot P)$. For k clients, the time complexity of gradient computation is $\mathcal{O}(E \cdot P \cdot k)$.
- *The computation of the ℓ_2-norm of the gradient.* Calculating the ℓ_2-norm of the gradient involves summing the squares of each component of the gradient vector g, followed by taking the square root. Therefore, the complexity of calculating $\|g\|_2$ is $\mathcal{O}(P)$, as it requires iterating through all P gradient components to compute the sum of squares. For k clients, The time complexity of this step is $\mathcal{O}(P \cdot k)$.
- *Comparison with the threshold C and clipping.* After calculating $\|g\|_2$, a comparison with the threshold C is made. This comparison operation itself has a constant time complexity of $\mathcal{O}(1)$. If gradient clipping is required, the normalization operation $g' \leftarrow \frac{g}{\max(1, \frac{\|g\|_2}{C})}$ involves scaling each component of the vector once, and its complexity is $\mathcal{O}(P)$. For k clients, The time complexity of this step is $\mathcal{O}(P \cdot k)$.

Dynamic Clipping Thresholds Calculation. According to Section IV-B, the time complexity of calculating dynamic clipping thresholds consists of three steps.

- *Statistics of Client Comparison Results.* Each client needs to calculate the ℓ_2-norm of its gradient $\|g_t(i)\|_2$, with a time complexity of $\mathcal{O}(P)$, where P is the dimension of the gradient vector (i.e., the number of model parameters). Then, the current threshold C_t is compared, with an $\mathcal{O}(1)$ complexity. Therefore, for each client, the time complexity for counting the number of conditions met (i.e., Num_i^{low} and Num_i^{high}) is $\mathcal{O}(P)$. For k clients, the total time complexity of this step is $\mathcal{O}(P \cdot k)$.
- *Calculation of the Update Ratio $\overline{L_t'}$.* Calculating the update ratio $\overline{L_t'}$ involves summing the statistical results of all clients and then dividing by $2k$. As it requires iterating through the statistical values of all clients, the time complexity of this step is $\mathcal{O}(k)$.
- *Updating the Clipping Threshold C_{t+1}.* The operation of updating the threshold C_{t+1} mainly involves using exponential functions and multiplication operations, which are constant-time operations with a complexity of $\mathcal{O}(1)$.

Combining all steps, we can conclude that the total time complexity of dynamically calculating the gradient clipping threshold C in each training round is $\mathcal{O}(P \cdot k)$. This complexity mainly arises from the calculation and comparison operations of the ℓ_2-norm of the gradients for all clients.

V. Performance Evaluation

A. Experimental Settings

Experimental Environment. The experimental environment consists of one parameter server and 100 clients. The

TABLE II
DATASETS DETAILS AND HYPERPARAMETER SETTINGS

Datasets	MNIST	CIFAR-10	CIFAR-100	Shakespeare
Type	Image	Image	Image	Text
Model	CNN	CNN	CNN	RNN
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

deep learning framework is PyTorch, and the Python version is 3.6. The compute nodes run on a 64-bit Ubuntu 20.04 LTS operating system. The CPU is an Intel(R) Xeon(R) Gold 6326 @2.90GHz, with 256GB of RAM, and a 4TB hard drive. The GPU is an NVIDIA $A100$ with 80GB of memory.

Non-IID Datasets and Target Models. The experiments are conducted on three image datasets (MNIST, CIFAR-10, and CIFAR-100) and one text dataset (Shakespeare dataset). Two target models (CNN and RNN) are trained on these datasets. Since research conducted under non-independent and identically distributed (non-IID) settings is common and more closely reflects real-world scenarios, we use the *Dirichlet* function $Dir(\varphi = 1)$ to partition the datasets [25, 27, 34], generating non-IID FL training datasets for different clients. Note that when using the *Dirichlet* function $Dir(\varphi)$ to generate non-IID datasets, the higher the value of the parameter φ, the more similar the distribution of training datasets allocated to different clients. Therefore, to more accurately simulate the real-world environment for models, we choose $\varphi = 1$ to create the non-IID settings required for our experiments.

- MNIST [23]. We train a Convolutional Neural Network (CNN) for image classification tasks, which consists of 2 convolutional layers (5×5, each activated by ReLU and followed by 2×2 max pooling), 2 fully connected layers and Softmax normalizes the final output.
- CIFAR-10 and CIFAR-100 [24]. We also train a CNN for image classification tasks, consisting of 3 fully connected layers, and the other settings are the same as MNIST.
- Shakespeare dataset is constructed from *The Complete Works of William Shakespeare* [26]. We follow the same settings in [25] to process the raw data, each client is assigned one or more lines for training or testing. We utilize a Recurrent Neural Network (RNN) to predict the next character. The RNN accepts an input sequence of 80 characters and includes an embedding layer (80×8), two LSTM layers (80×256), and a dense layer (80×90).

Baselines. Note that both DP-FedAGNC and DP-FedDDC can achieve dynamic gradient clipping.

- FedAvg randomly selects a subset of clients to participate in each round of FL training and averages the gradients uploaded by the clients [26]. FedAvg is already the most commonly used and classic FL baseline method without privacy consideration.
- DP-FedAvg is a common baseline method that introduces DP noise into FedAvg to enhance privacy protection [10].
- DP-FedAGNC utilizes the average ℓ_2-norm of gradients from the previous batch as the clipping threshold for the next batch [6].
- DP-FedDDC employs a near-linear decay function to set the clipping threshold and adaptively adjusts the noise scaler [18]. DP-FedDDC is the state-of-the-art gradient clipping method under higher privacy budgets.

Hyperparameter Settings. To ensure a fair comparison between DP-FedACN and baselines, we follow the same settings in DP-FedDDC and set the relaxation parameter $\delta = \frac{1}{10|D|} = 0.001$ [18], where $|D|$ is the training data sample size for each client. We set the privacy budgets $\varepsilon = \{0.1, 0.2, 0.5, 1, 2, 4\}$ to demonstrate the robustness of DP-FedACN. Each round of FL training randomly selects 10 clients to participate in the training. The other parameters follow common settings used in image classification tasks and next-character prediction tasks, with details provided in Table II.

Member Inference Attacks (MIAs) and Model Inversion (MI) Attack in Security Model. Basic-MIA, ML-Leaks, and White-box member inference attack methods are employed to perform inference attacks during FL training [20–22]. We use equal-sized sets to ensure an equal number of members and non-members, in order to maximize the uncertainty of the inference. Additionally, to evaluate the privacy protection performance of DP-FedACN against other potential attacks or more complex privacy leakage scenarios, Knowledge-Enriched Distributional Model Inversion attack (KED-MI) [35], which provides state-of-the-art performance for white-box MI attacks, is employed to perform model inversion attacks.

- *ML-Leaks (Adversary 1):* The adversary initially splits the shadow dataset D_{shadow} into two subsets, namely D^{train}_{shadow} and D^{test}_{shadow}. Subsequently, a shadow model M_{shadow} is trained using the data from D^{train}_{shadow}. From the output of M_{shadow}, the adversary identifies the three highest posterior values and assigns them labels of either 1 or 0. Finally, the adversary generates predictions related to membership status.
- *White-box Inference:* Adversaries train their adversarial models using diverse components from both training and testing datasets. By targeting multiple observed inputs of the target model, the adversary captures correlations between parameters across different iteration rounds.
- *KED-MI Attack:* In white-box MI attacks, with access to a target model $M^d \rightarrow \mathbb{R}^{|C|}$ and any given target class $c^* \in C$, the objective of the adversary is to reconstruct a feature point x^* from the training data associated with class c^*. d is the dimension of the model input, C is the set of all class labels, and $|C|$ is the size of C. Following the experimental setup in [35], we evaluate KED-MI's attack performance using the same attack accuracy metric (called attack success rate in this paper). A high attack success rate indicates that the reconstructed images may reveal private information about the target label.

Metrics. We use the four evaluation criteria.

- *Privacy Protection.* We utilize three MIAs and one MI attack to evaluate the privacy protection performance of the four methods. The lower the attack success rate

TABLE III
ATTACK SUCCESS RATE OF DIFFERENT ATTACK METHODS WITH DIFFERENT TRAINING METHODS (ASR%)

Privacy Budgets	Methods (DP-)	MNIST				CIFAR-10				CIFAR-100			
		Basic-MIA	ML-Leaks	White-box	KED-MI	Basic-MIA	ML-Leaks	White-box	KED-MI	Basic-MIA	ML-Leaks	White-box	KED-MI
1	FedAvg	50.13%	50.56%	51.35%	51.48%	58.43%	62.01%	67.84%	59.20%	62.83%	69.01%	74.44%	63.32%
	FedAGNC	50.39%	50.81%	51.62%	51.91%	59.05%	63.03%	68.30%	60.53%	63.16%	75.45 %	78.29%	63.55%
	FedDDC	50.42%	50.63%	52.04%	52.27%	59.08%	62.47%	68.38%	60.61%	63.24%	69.88%	78.86%	63.61%
	FedACN	**50.47%**	**50.61%**	**51.38%**	**51.53%**	**59.19%**	**62.42%**	**68.26%**	**59.86%**	**63.31%**	**69.36%**	**75.77%**	**63.38%**
2	FedAvg	50.22%	50.69%	51.64%	51.55%	59.41%	63.11%	70.95%	60.15%	64.51%	75.09%	77.99%	65.00%
	FedAGNC	50.44%	50.96%	51.81%	52.14%	60.03%	64.06%	71.12%	61.09%	64.86%	76.82%	80.20%	65.69%
	FedDDC	50.48%	50.78%	51.98%	52.10%	60.06%	63.68%	71.20%	61.14%	64.90%	75.30%	80.39%	65.66%
	FedACN	**50.52%**	**50.77%**	**51.77%**	**52.01%**	**60.16%**	**63.47%**	**71.07%**	**60.91%**	**65.02%**	**75.24%**	**78.11%**	**65.35%**
4	FedAvg	50.29%	50.89%	52.06%	52.62%	60.45%	63.54%	76.02%	61.30%	68.25%	78.59%	80.95%	66.12%
	FedAGNC	50.51%	51.04%	52.11%	53.22%	61.17%	65.12%	75.88%	61.74%	68.87%	84.61%	85.32%	67.34%
	FedDDC	50.56%	50.98%	52.24%	53.28%	61.22%	64.21%	76.44%	61.66%	68.94%	79.07%	86.08%	67.40%
	FedACN	**50.60%**	**50.96%**	**52.07%**	**53.15%**	**61.31%**	**63.68%**	**76.34%**	**61.39%**	**69.11%**	**78.95%**	**82.46%**	**66.91%**

TABLE IV
AVERAGE ATTACK SUCCESS RATE OF THREE MIAS WITH DIFFERENT TRAINING METHODS (AASR%)

Privacy Budgets	Methods (DP-)	MNIST	CIFAR-10	CIFAR-100
1	FedAvg	50.68%	62.76%	68.76%
	FedAGNC	50.94%	63.46%	72.30%
	FedDDC	51.03%	63.31%	70.66%
	FedACN	**50.82%**	**63.29%**	**69.48%**
2	FedAvg	50.85%	64.49%	72.53%
	FedAGNC	51.07%	65.07%	73.96%
	FedDDC	51.08%	64.98%	73.53%
	FedACN	**51.02%**	**64.90%**	**72.79%**
4	FedAvg	51.08%	66.67%	75.93%
	FedAGNC	51.22%	67.39%	79.60%
	FedDDC	51.26%	67.29%	78.03%
	FedACN	**51.21%**	**67.11%**	**76.84%**

TABLE V
GLOBAL AVERAGE TEST ACCURACY WITH DIFFERENT PRIVACY BUDGETS FOR DIFFERENT TRAINING METHODS (ATA%)

Privacy Budgets	Methods	MNIST	CIFAR-10	CIFAR-100	Shakespeare
-	FedAvg	98.84%	73.25%	38.62%	62.74%
1	DP-FedAvg	95.12%	52.73%	22.74%	34.15%
	DP-FedAGNC	95.72%	54.46%	24.52%	36.80%
	DP-FedDDC	95.61%	54.17%	24.09%	36.38%
	DP-FedACN	**95.94%**	**55.74%**	**25.81%**	**36.95%**
2	DP-FedAvg	96.27%	55.87%	24.97%	39.61%
	DP-FedAGNC	96.62%	57.69%	26.77%	41.53%
	DP-FedDDC	96.58%	57.61%	26.62%	41.66%
	DP-FedACN	**97.02%**	**59.30%**	**28.15%**	**42.49%**
4	DP-FedAvg	96.89%	58.10%	26.53%	44.97%
	DP-FedAGNC	97.21%	60.27%	28.45%	47.17%
	DP-FedDDC	97.37%	60.72%	28.76%	47.43%
	DP-FedACN	**97.87%**	**61.80%**	**29.74%**	**48.52%**

(ASR%) and average attack success rate (AASR%), the higher the privacy protection performance.

- *Global Model Availability.* We employ the global average testing accuracy (ATA%) of model training to evaluate the global model availability of DP-FedACN. Note that a higher global average test accuracy (ATA%) in the experimental results indicates better global model availability.
- *Clipping Threshold Universality.* Similarly to existing methods, we measure the threshold universality using the standard deviation of accuracy. The accuracy standard deviation $SD = \sqrt{\frac{1}{i}\sum(A_i - \overline{A})^2}$ is the standard deviation between the accuracy A_i of the global model on each local client's test dataset and the average accuracy $\overline{A}$ across all clients. A smaller SD indicates that in each iteration, the clipping threshold is closer to the optimal value, suggesting more reasonable threshold adjustments and greater universality.
- *Applicability of DP-FedACN.* We compare the global average testing accuracy (ATA%) of model training at lower privacy budgets $\varepsilon = \{0.1, 0.2, 0.5\}$ to evaluate the applicability of DP-FedACN. Note that a higher average test accuracy (ATA%) in the experimental results indicates better applicability.

B. Privacy Protection

One of the most direct methods to evaluate privacy protection performance is incorporating inference attacks during the model training. In this section, we conduct comparative experiments using three MIAs (i.e., Basic-MIA, ML-Leaks, and White-box attacks) and one MI attack (KED-MI). We evaluate the privacy protection performance of the four DP methods by analyzing the attack success rate (ASR%) under different attacks. We train a global model on MNIST and CIFAR-10/100 datasets using DP-FedAvg, DP-FedAGNC, DP-FedDDC, and DP-FedACN with the privacy budgets $\varepsilon = \{1, 2, 4\}$. The experimental results are shown in Table III.

1) Training with Different ε and MIAs. DP-FedAvg consistently exhibits the lowest ASR and has superior privacy protection performance. This is because DP-FedAvg only prevents the addition of substantial noise and clip gradients using a fixed threshold. Since DP-FedACN can clip the gradient more accurately in each training round and introduce fewer noise perturbations, DP-FedACN has the highest ASR when facing Basic-MIA. However, when attacked by stronger MIAs (i.e., ML-Leaks and White-box attacks), the ASR of DP-FedACN is much lower than that of DP-FedAGNC and DP-FedDDC. From the experimental results reported in Table III, the privacy protection performance of DP-FedACN is similar to that of DP-FedAvg and outperforms DP-FedAGNC and DP-FedDDC. This is because DP-FedACN adjusts the clipping threshold

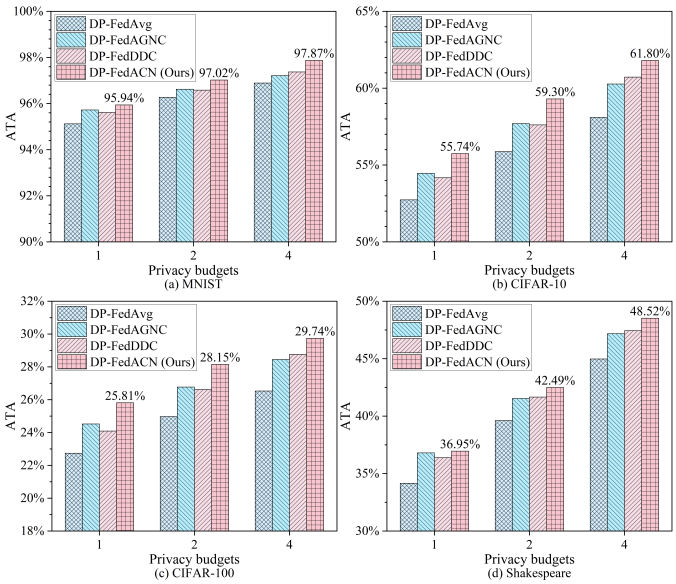


Fig. 4. Global average test accuracy on four datasets (ATA%).

TABLE VI
GLOBAL AVERAGE TEST ACCURACY ON FOUR DATASETS (ATA%)

Datasets	**DP-FedACN**	DP-FedAvg	DP-FedAGNC	DP-FedDDC
MNIST	**96.94%**	96.09% (**-0.85**)	96.52% (**-0.42**)	96.52% (**-0.42**)
CIFAR-10	**58.95%**	55.57% (**-3.3**)	57.47% (**-1.48**)	57.50% (**-1.45**)
CIFAR-100	**27.90%**	24.75% (**-3.15**)	26.58% (**-1.32**)	26.49% (**-1.41**)
Shakespeare	**42.65%**	39.58% (**-3.07**)	41.83% (**-0.82**)	41.82% (**-0.83**)
Average	**56.61%**	54.00% (**-2.61**)	55.60% (**-1.01**)	55.58% (**-1.03**)

by considering the gradient norm changes and uses a decay function to control the trend of clipping threshold changes.

2) Training with Different ε and MI attacks. When training with the MNIST dataset, the ASR of MI attacks for the four methods is higher than MIAs. However, when training with the CIFAR-10/100 datasets, the ASR of MI attacks for the four methods is lower than MIAs. This is because the complexity and diversity of the CIFAR-10/100 datasets make it very challenging to reverse-engineer a high-quality image that can be correctly classified. Under different privacy budgets $\varepsilon = \{1, 2, 4\}$, the ASR of DP-FedACN is slightly higher than DP-FedAvg, but significantly lower than DP-FedAGNC and DP-FedDDC. Therefore, when facing MI attacks, the privacy protection performance of DP-FedACN is comparable to DP-FedAvg and outperforms DP-FedAGNC and DP-FedDDC.

3) Training with Different ε and Datasets. As shown in Table IV, we calculate the AASR of the three MIAs. When training with the MNIST dataset, the AASR of the four methods does not differ significantly. However, when training with the high-complexity CIFAR-10/100 datasets, the AASR of the four methods sharply increases. The AASR of DP-FedACN is lower than that of DP-FedAGNC and DP-FedDDC. Particularly when training with high-complexity datasets and high privacy budgets, DP-FedACN exhibits superior privacy protection compared to DP-FedAGNC and DP-FedDDC but is slightly inferior to DP-FedAvg by less than 1%.

4) Summary. DP-FedACN shows better defense against MIAs and MI attacks under high privacy budgets. The privacy protection performance of DP-FedACN is comparable to DP-FedAvg and outperforms DP-FedAGNC and DP-FedDDC. In summary, the more complex the data used for model training and the stronger the MIAs, the better the privacy protection performance of DP-FedACN.

C. Global Model Availability

In this section, we use global average testing accuracy (ATA%) to evaluate the global model availability of DP-FedACN. Table V and Fig. 4 illustrate the changes in ATA for DP-FedAvg, DP-FedAGNC, DP-FedDDC, and DP-FedACN with the privacy budgets $\varepsilon = \{1, 2, 4\}$.

1) Training with Different ε. For the MNIST dataset, when the privacy budget $\varepsilon = 1$, the ATA of DP-FedACN is higher than that of DP-FedAvg by approximately 0.82%. When $\varepsilon = 4$, the ATA of DP-FedACN is higher than that of DP-FedAvg by approximately 0.98%. Similarly, for the CIFAR-10/100 datasets, when $\varepsilon = 4$, the ATA of DP-FedACN is higher than that of DP-FedAvg by approximately 3.7% and 3.21%, respectively. These improvements are higher than the ones at $\varepsilon = 1$, which are 3.01% and 3.07%, respectively. Therefore, DP-FedACN can better improve the global model's availability under high privacy budgets. For the Shakespeare dataset, DP-FedACN achieves a 3.55% higher ATA than DP-FedAvg with $\varepsilon = 4$, and a 2.80% higher ATA than DP-FedAvg with $\varepsilon = 1$, which is consistent with the experimental findings on the MNIST and CIFAR-10/100 datasets. This demonstrates that DP-FedACN provides a greater improvement in model accuracy with higher privacy budgets compared to lower privacy budgets when training on text datasets. In addition, DP-FedACN always achieves significantly higher ATA when $\varepsilon = \{1, 2, 4\}$ compared to DP-FedAvg, DP-FedAGNC, and DP-FedDDC. This indicates that FedACN can also achieve better model accuracy on text datasets.

2) Training with Different Datasets. When training with the relatively simple MNIST dataset, the differences in ATA among the four methods are not significant. However, when training with the complex CIFAR-10/100 and Shakespeare datasets, DP-FedACN exhibits significantly higher ATA compared to DP-FedAvg, DP-FedAGNC, and DP-FedDDC. This is because when training with complex datasets, the clipping threshold in each training round needs to be more precise to prevent adding excessive noise. DP-FedACN can capture the overall trend of gradient norm changes and incorporate these changes into the control mechanism for dynamically adjusting the clipping threshold in each training round. Therefore, the clipping threshold dynamically adjusted by DP-FedACN will be more precise and adaptive to the changes in gradient norms.

3) Summary. Table VI shows the ATA of the four methods on four different datasets. For the MNIST, CIFAR-10/100 and Shakespeare datasets, DP-FedACN achieves higher ATA compared to the DP-FedAvg, DP-FedAGNC, and DP-FedDDC

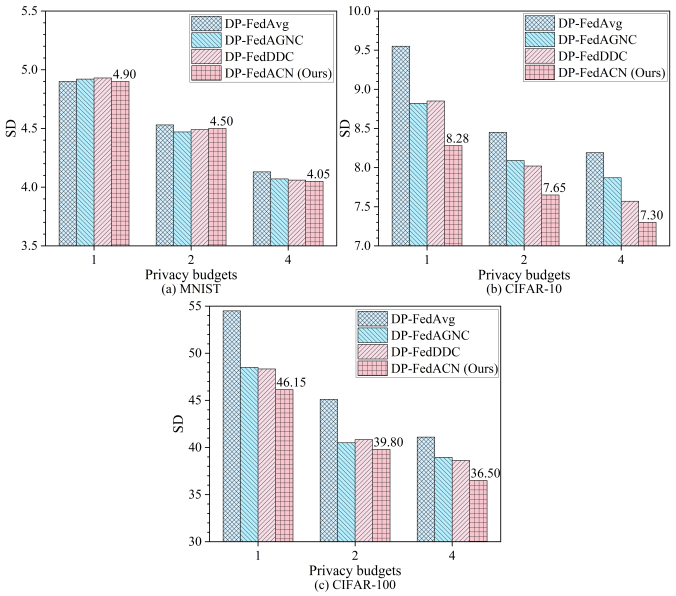


Fig. 5. Local test accuracy standard deviation on image datasets (SD).

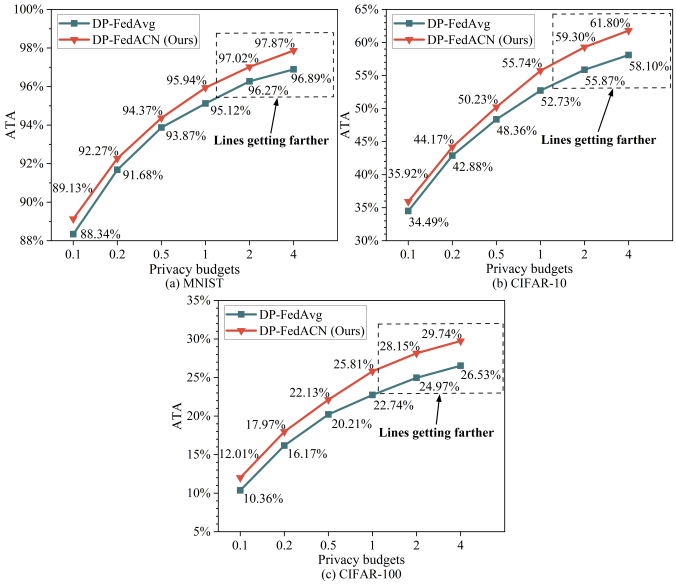


Fig. 6. Global average test accuracy on image datasets with lower privacy budgets $\varepsilon = \{0.1, 0.2, 0.5\}$ (ATA%).

methods by approximately 2.61%, 1.01%, and 1.03%, respectively. The global model availability of DP-FedACN is superior. Especially under high privacy budgets, the ATA of DP-FedACN significantly outperforms the other three methods. Despite the fact that DP-FedACN achieves only 1.45% and 1.03% higher model accuracy on image and text datasets respectively compared to the state-of-the-art dynamic clipping method DP-FedDDC, experimental results in Section V-B show that the ASR of DP-FedACN is close to DP-FedAvg and significantly is lower than DP-FedDDC (the largest gap is 1.19%, $\varepsilon = 4$, CIFAR-100). These improvements indicate that DP-FedACN can provide better privacy protection and model accuracy compared to DP-FedDDC, thereby exploring a better trade-off between model privacy and accuracy.

D. Clipping Threshold Universality

We utilize the standard deviation (SD) of local testing accuracy to evaluate the universality of the clipping thresholds calculated using DP-FedACN. Fig. 5 illustrates the standard deviation for DP-FedAvg, DP-FedAGNC, DP-FedDDC, and DP-FedACN with the privacy budgets $\varepsilon = \{1, 2, 4\}$.

As shown in Fig. 5, when training on the MNIST dataset, the differences in local testing accuracy standard deviation among the four methods are not significant, as the MNIST dataset has relatively low complexity. However, when training on the more complex CIFAR-10/100 datasets, DP-FedACN exhibits lower local testing accuracy standard deviation under all three privacy budgets compared to DP-FedAvg, DP-FedAGNC, and DP-FedDDC. This is because DP-FedAvg adopts a fixed threshold for clipping, leading to the poorest gradient clipping effect. On one hand, DP-FedACN considers the relationship between the clipping threshold and the gradient norm. When adjusting the clipping threshold in each training round, DP-FedACN incorporates information about the changes in gradient norms into the adjustment method. On the other hand, throughout the entire training process, DP-FedACN macroscopically controls the future trend of the clipping threshold. This avoids potential misdirection of the threshold due to gradient norm fluctuations in a specific training round. These improvements allow DP-FedACN to achieve higher precision and effectiveness in gradient clipping. In summary, the universality of the clipping thresholds in DP-FedACN is better than that of the other three methods.

E. Applicability Analysis under Lower Privacy Budgets

Section V-B shows that the privacy protection performance of DP-FedACN is comparable to that of DP-FedAvg and outperforms DP-FedAGNC and DP-FedDDC. Section V-C shows that the global model availability of DP-FedACN is the best especially when training with high privacy budgets. In this section, we choose DP-FedAvg as the comparative method to evaluate the performance of DP-FedACN under low privacy budgets $\varepsilon = \{0.1, 0.2, 0.5\}$ by comparing the average testing accuracy (ATA%) of DP-FedACN and DP-FedAvg.

As shown in Fig. 6, DP-FedACN consistently achieves higher ATA than DP-FedAvg. It is noteworthy that as the privacy budget increases, the increment in ATA for DP-FedACN gradually surpasses that of DP-FedAvg. This is because DP-FedACN adds noise after clipping gradients. Although the amount of noise added can be dynamically adjusted by considering the clipping threshold changes, the impact of the privacy budget on the perturbation of gradients remains significant. This leads to a decrease in accuracy for DP-FedACN under low privacy budgets. As the privacy budget becomes larger, the influence of the noise added on accuracy diminishes, and DP-FedACN can dynamically adjust the clipping threshold and choose an appropriate noise distribution for the current iteration round, making the model training more accurate. Therefore, when training with low privacy budgets, DP-FedACN can still maintain a higher ATA than DP-FedAvg. As the privacy budget gradually increases, the

superiority of DP-FedAvg becomes more evident, and DP-FedACN can achieve significantly higher ATA. Thus, DP-FedAvg demonstrates good universality.

VI. Conclusion

Fixed or imprecise thresholds are not adaptive to the changes in gradients. This shortcoming can lead to excessive noise addition and significantly degradate model accuracy. To address this issue, we propose Differential Privacy Federated Adaptive gradient Clipping based on gradient Norm (DP-FedACN). In each training round, DP-FedACN calculates the decay rate of the clipping threshold by capturing the overall trend of gradient norm changes. Thus, DP-FedACN can calculate an adaptive clipping threshold by considering multiple factors such as the changes in gradient norms, clipping loss, and decay rate. Experimental results demonstrate that DP-FedACN achieves higher average testing accuracy compared to the three baseline methods by approximately 2.61%, 1.01%, and 1.03%, respectively. DP-FedACN can accurately adjust the gradient clipping threshold in each training round. Therefore, DP-FedACN allows precise control over the amount of added noise and effectively improves the accuracy of global model training. In summary, DP-FedACN can help find a fine-grained privacy-accuracy trade-off for DP-FL.

References

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM Trans. Intell. Syst. Technol.* 2019.

[2] 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.

[3] X. Huang, Y. Ding, Z. L. Jiang, S. Qi, X. Wang, and Q. Liao, "DP-FL: A Novel Differentially Private Federated Learning Framework for The Unbalanced Data," *World Wide Web*. 2020.

[4] A. E. Ouadrhiri and A. Abdelhadi, "Differential Privacy for Deep and Federated Learning: A Survey," in *IEEE Access*. 2022.

[5] H. Yang, M. Ge, D. Xue, K. Xiang, H. Li, and R. Lu, "Gradient Leakage Attacks in Federated Learning: Research Frontiers, Taxonomy and Future Directions," *IEEE Network*. 2023.

[6] K. L. van der Veen, R. Seggers, P. Bloem, and G. Patrini, "Three Tools for Practical Differential Privacy," in *Proceedings of the NeurIPS 2018 Workshop*. 2018.

[7] K. Wei *et al*., "Federated Learning With Differential Privacy: Algorithms and Performance Analysis," in *IEEE Transactions on Information Forensics and Security*. 2020.

[8] D. Yu, "Improve the Gradient Perturbation Approach for Differentially Private Optimization," in *Proceedings of the NeurIPS 2018 Workshop*. 2018.

[9] Y. Yuan, Z. Zou, D. Li, L. Yan, and D. Yu, "D-(DP)2SGD: Decentralized Parallel SGD with Differential Privacy in Dynamic Networks," *Wireless Communications and Mobile Computing*. 2021.

[10] 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*. 2016.

[11] Z. Pan, L. Hu, W. Tang, J. Li, Y. He and Z. Liu, "Privacy-Preserving Multi-Granular Federated Neural Architecture Search–A General Framework," in *IEEE Transactions on Knowledge and Data Engineering*. 2023.

[12] W. Tang, B. Li, M. Barni, J. Li and J. Huang, "An Automatic Cost Learning Framework for Image Steganography Using Deep Reinforcement Learning," in *IEEE Transactions on Information Forensics and Security*. 2021.

[13] T. Li, J. Li, X. Chen, Z. Liu, W. Lou, and Y. T. Hou, "NPMML: A Framework for Non-Interactive Privacy-Preserving Multi-Party Machine Learning," in *IEEE Transactions on Dependable and Secure Computing*. 2021.

[14] H. Liu, C. Li, B. Liu, P. Wang, S. Ge, and W. Wang, "Differentially Private Learning with Grouped Gradient Clipping," in *Proceedings of the 3rd ACM International Conference on Multimedia in Asia*. 2022.

[15] Y. Guo, Q. Wang, T. Ji, X. Wang and P. Li, "Resisting Distributed Backdoor Attacks in Federated Learning: A Dynamic Norm Clipping Approach," *2021 IEEE International Conference on Big Data (Big Data)*. 2021.

[16] N. Wang, Y. Xiao, Y. Chen, N. Zhang, W. Lou, and Y. T. Hou, "Squeezing More Utility via Adaptive Clipping on Differentially Private Gradients in Federated Meta-Learning," in *Proceedings of the 38th Annual Computer Security Applications Conference*, in ACSAC '22. 2022.

[17] R. Ramakrishna, A. Scaglione, T. Wu, N. Ravi, and S. Peisert, "Differential Privacy for Class-Based Data: A Practical Gaussian Mechanism," in *IEEE Transactions on Information Forensics and Security*. 2023.

[18] J. Du, S. Li, X. Chen, S. Chen, and M. Hong, "Dynamic Differential-Privacy Preserving SGD." 2021.

[19] 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,* 2017.

[20] 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*. 2022.

[21] 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.

[22] 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*. 2019.

[23] Y. LeCun and C. Cortes, "MNIST Handwritten Digit

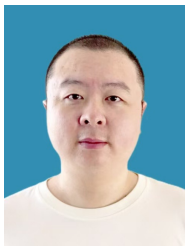
Database." 2010.
[24] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," Dept. Comput. 2009.
[25] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konecny, S. Kumar, and H. B. McMahan, "Adaptive Federated Optimization." 2020.
[26] B. McMahan, E. Moore, D. Ramage, S. Hampson, 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*. 2017.
[27] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification." 2019.
[28] T. Hoang, A. A. Yavuz, and J. G. Merchan, "A Secure Searchable Encryption Framework for Privacy-Critical Cloud Storage Services," *IEEE Trans. Services Comput*. 2021.
[29] 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*. 2016.
[30] Z. Tang *et al.*, "FedImpro: Measuring and Improving Client Update in Federated Learning," In *ICLR*. 2024.
[31] Z. Tang, Y. Zhang, S. Shi, X. He, B. Han, and X. Chu, "Virtual Homogeneity Learning: Defending against Data Heterogeneity in Federated Learning," in *Proceedings of the 39th International Conference on Machine Learning, PMLR*. 2022.
[32] Z. Tang, S. Shi, B. Li and X. Chu, "GossipFL: A Decentralized Federated Learning Framework With Sparsified and Adaptive Communication," in *IEEE Transactions on Parallel and Distributed Systems*. 2023.
[33] J. Li *et al.*, "Blockchain Assisted Decentralized Federated Learning (BLADE-FL): Performance Analysis and Resource Allocation," in *IEEE Transactions on Parallel and Distributed Systems*. 2022.
[34] Y. Shi, Y. Liu, K. Wei, L. Shen, X. Wang, and D. Tao, "Make Landscape Flatter in Differentially Private Federated Learning," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.
[35] S. Chen, M. Kahla, R. Jia, and G.-J. Qi, "Knowledge-Enriched Distributional Model Inversion Attacks," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.
[36] R. C. Geyer, T. Klein, and M. Nabi, "Differentially Private Federated Learning: A Client Level Perspective." *arXiv*, Mar. 01, 2018.
[37] G. Andrew, O. Thakkar, H. B. McMahan, and S. Ramaswamy, "Differentially Private Learning with Adaptive Clipping," in *Proceedings of the 35th International Conference on Neural Information Processing Systems (NIPS)*, pp. 17455–17466, 2021.

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 pursuing 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.

Yingchi Mao (Member, IEEE) received the Ph.D. degree in computer software and theory from the Department of Computer Science and Technology, Nanjing University, Nanjing, China in 2007. She serves with the Key Laboratory of Water Big Data Technology, Ministry of Water Resources, Nanjing, and she is also currently a Professor with the College of Computer Science and Software Engineering, Hohai University, Nanjing. Her main research interests include edge intelligent computing, Internet of Things data analysis, and mobile sensing systems.

Prof. Mao is a Senior Member of the China Computer Federation and the 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 in Singapore University of Technology and Design.

His current research interests include edge intelligence and FPGA-based AI accelerators.

Ping Ping (Member, IEEE) received the Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2009. She is currently an Assistant Professor with the College of Computer Science and Software Engineering, Hohai University, Nanjing.

Her research interests include network and information security, cloud computing and big data security, and image-hiding encryption.

Huawei Huang (M'16-SM'22) received his Ph.D. degree in Computer Science and Engineering from the University of Aizu (Japan) in 2016. He is currently an Associate Professor at 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 JSAC and IEEE OJ-CS.

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 Temple University, Philadelphia, PA, USA, and also serves as the Director of International Affairs, College of Science and Technology. Dr. Wu 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.