

Article

Differentially Private Federated Learning with Adaptive Clipping Thresholds

Jianhua Liu , Yanglin Zeng , Zhongmei Wang *, Weiqing Zhang and Yao Tong 

School of Transportation and Electrical Engineering, Hunan University of Technology, Zhuzhou 412007, China; jhliu@hut.edu.cn (J.L.); m23085400064@stu.hut.edu.cn (Y.Z.); m23081100001@stu.hut.edu.cn (W.Z.); tongyao@stu.hut.edu.cn (Y.T.)

* Correspondence: wangzhongmei@hut.edu.cn

Abstract

Under non-independent and identically distributed (Non-IID) conditions, significant variations exist in local model updates across clients and training phases during the collaborative modeling process of differential privacy federated learning (DP-FL). Fixed clipping thresholds and noise scales struggle to accommodate these diverse update differences, leading to mismatches between local update intensity and noise perturbations. This imbalance results in data privacy leaks and suboptimal model accuracy. To address this, we propose a differential privacy federated learning method based on adaptive clipping thresholds. During each communication round, the server adaptively estimates the global clipping threshold for that round using a quantile strategy based on the statistical distribution of client update norms. Simultaneously, clients adaptively adjust their noise scales according to the clipping threshold magnitude, enabling dynamic matching of clipping intensity and noise perturbation across training phases and clients. The novelty of this work lies in a quantile-driven, round-wise global clipping adaptation that synchronizes sensitivity bounding and noise calibration across heterogeneous clients, enabling improved privacy–utility behavior under a fixed privacy accountant. Using experimental results on the rail damage datasets, our proposed method slightly reduces the attacker’s MIA ROC-AUC by 0.0033 and 0.0080 compared with Fed-DPA and DP-FedAvg, respectively, indicating stronger privacy protection, while improving average accuracy by 1.55% and 3.35% and achieving faster, more stable convergence. We further validate its effectiveness on CIFAR-10 under non-IID partitions.

Keywords: federated learning; client-level differential privacy; adaptive clipping; quantile statistics; RDP accounting; membership inference attack



Academic Editor: Carlo Blundo

Received: 6 February 2026

Revised: 5 March 2026

Accepted: 12 March 2026

Published: 14 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Federated learning (FL) enables multiple parties to collaboratively train a shared model under the principle of “data stays local, only parameters travel” [1–3]. This paradigm has attracted increasing attention in privacy-sensitive applications such as medical diagnosis, financial risk control, and industrial monitoring, where data centralization is restricted. Nevertheless, keeping raw data local does not automatically eliminate privacy risks: gradients or model updates uploaded by clients may still encode statistical information about local training data, which can be exploited by adversaries to infer whether specific samples participated in training (membership inference) and further reveal sensitive attributes or distributional characteristics of participants [4–9]. Such leakage poses serious threats to

individual privacy and institutional data security. Therefore, providing quantifiable and controllable privacy protection while maintaining model utility remains a critical challenge for real-world FL deployment.

Differential privacy (DP) has been widely incorporated into FL as a principled mechanism that provides formal privacy guarantees and an explicit privacy–utility trade-off [10–12]. In typical client-level DP-FL, each client update is norm-clipped to bound sensitivity, and Gaussian noise is injected before aggregation. Privacy accounting mechanisms (e.g., Rényi differential privacy, RDP) are further used to track the accumulated privacy loss across communication rounds [11,13]. Despite its practicality, many DP-FL deployments still rely on fixed global clipping thresholds. When client update magnitudes are relatively homogeneous, a fixed threshold may offer acceptable training stability. However, under pronounced Non-IID heterogeneity, update magnitudes and convergence stages can vary substantially across clients and training phases, making a single static threshold difficult to accommodate. Specifically, an overly small threshold may over-compress many client updates, causing excessive information loss and slowing convergence; an overly large threshold increases the effective clipping threshold, allowing a few large updates to exert disproportionate influence after clipping and potentially inducing aggregation instability or bias. Moreover, because the noise magnitude commonly scales with the clipping threshold, an excessively large threshold introduces redundant noise, which may drown out small-magnitude updates—especially in later training stages—thereby degrading model utility and complicating the privacy–utility balance.

To address these challenges, we propose AdaCT-DPFL, a differentially private federated learning method with adaptive clipping thresholds. AdaCT-DPFL estimates the global clipping threshold at the server side in each communication round via a quantile-based strategy over the distribution of client update norms, enabling the clipping strength to adapt to heterogeneous clients and evolving training phases. Meanwhile, the noise scale is adjusted in accordance with the estimated threshold, alleviating both over-clipping and noise redundancy induced by fixed thresholds. As a result, AdaCT-DPFL improves convergence stability and final performance under identical privacy constraints, and enhances robustness against membership inference attacks.

Contributions

- We identify a key mismatch in client-level DP-FL under Non-IID heterogeneity: fixed clipping can simultaneously cause over-clipping for many clients and redundant noise for small updates in later rounds.
- We propose AdaCT-DPFL, which estimates a round-wise global clipping threshold via quantile statistics of client update norms and synchronizes Gaussian noise calibration accordingly.
- We clarify privacy accounting under adaptive thresholds using the RDP accountant, explaining how Clipping Thresholds vary under proportional noise calibration.
- We validate the method on CIFAR-10 and a real-world rail defect dataset with black-box membership inference attacks using multiple attack classifiers.

2. Related Work

Although FL keeps raw data local, prior studies have shown that gradients or model updates can still leak information about local training distributions, leading to practical privacy risks in federated deployments [4–8]. Membership inference attacks are commonly used to empirically assess such leakage in both centralized and federated settings [5,9], thereby motivating privacy mechanisms that provide rigorous protection beyond data locality.

To achieve formal and quantifiable privacy guarantees, DP has been extensively adopted in FL, establishing an explicit privacy–utility trade-off with principled accounting of privacy loss across training rounds [10–12]. In client-level DP-FL, sensitivity is typically bounded by applying norm clipping to each client update, followed by Gaussian perturbation prior to aggregation, and cumulative privacy consumption is tracked by standard accounting techniques such as RDP [11–13]. Recent DP-FL defenses further explore mitigating inference attacks using randomized response or related randomization mechanisms under DP settings [14]. Beyond the classical DP-FL pipeline, prior work has explored system- and algorithm-level designs for privacy-preserving distributed optimization, and surveys further summarize design choices and practical considerations for privacy-preserving machine learning, including cryptographic approaches and their challenges [12,15–17].

A central challenge in DP-FL is mitigating the utility degradation induced by clipping and noise. Existing studies have proposed improvements from multiple angles, including adaptive clipping and local training refinement to better match heterogeneous update scales, adaptive local-DP formulations, and mechanisms designed to improve robustness and performance under heterogeneous or edge environments [18–20]. Other work targets realistic training issues such as imbalance and evolving dynamics, proposing training adjustments under DP constraints to alleviate performance loss [21,22]. Moreover, dynamic or personalized federated learning with adaptive DP has been studied to better handle client heterogeneity by adapting learning behavior or privacy configurations across clients and stages [23]. In parallel, research on differentially private optimization investigates optimization-aware strategies (e.g., adaptive preconditioning) under privacy constraints, providing insights into the interaction between DP noise and optimizer dynamics [24]. Among these adaptive methods, DP-FedPUAC [18] adapts clipping on the client side by adjusting the clipping bound together with local iteration optimization, while AdaClip-style approaches similarly adapt clipping using heuristic or per-client statistics to stabilize DP-SGD/DP-FL training [25]. In contrast, our method estimates a round-wise global clipping threshold using quantile statistics over participating clients' update norms and synchronizes Gaussian noise calibration with the round-wise threshold, which differs from adaptive DP optimization (e.g., adaptive preconditioning [24]) that mainly modifies optimizer dynamics under DP constraints rather than performing round-wise global threshold estimation with matched noise calibration.

Nevertheless, many practical DP-FL deployments still adopt fixed global clipping thresholds or static noise calibration. Under Non-IID heterogeneity and time-varying training dynamics, static thresholds may over-clip a large portion of clients while allowing a few large updates to exert disproportionate influence after clipping. Furthermore, when the noise scale is coupled with the clipping threshold, static threshold choices may inject redundant perturbations during phases where most updates are small, suppressing effective learning signals and undermining the privacy–utility balance. These limitations motivate globally consistent adaptive thresholding mechanisms that can track the evolving distribution of client update magnitudes.

3. Problem Formulation

Let K denote the number of clients. Client k holds a local dataset D_k with $n_k = |D_k|$ samples, and the total number of samples is $N = \sum_{k=1}^K n_k$, the global model parameters at round t are denoted by $\theta^{(t)}$. After local training in round t , client k produces a model update $\Delta\theta_k^{(t)}$, and its L_2 -norm is $r_k^{(t)} = \|\Delta\theta_k^{(t)}\|_2$. The clipping threshold at round t is C_t . We use σ to denote the noise multiplier, so the Gaussian noise standard deviation is σC_t .

The local empirical risk of a client is defined as

$$f_k(\theta^{(t)}) = \frac{1}{n_k} \sum_{(x,y) \in D_k} \ell(\theta^{(t)}; x, y), \quad (1)$$

where $\ell(\theta; x, y)$ denotes the loss function. Client-level differentially private federated learning optimizes the global objective:

$$\min_{\theta} F(\theta^{(t)}) = \sum_{k=1}^K \frac{n_k}{N} f_k(\theta^{(t)}). \quad (2)$$

Differential Privacy (DP) [13,16,24] is a formal privacy framework providing theoretical guarantees against identifying private data within datasets. Within DP, ϵ is the privacy budget and δ is the upper bound on the failure probability. For any pair of adjacent datasets D and D' that differ in one record, and any measurable subset S of outputs, a randomized mechanism M satisfies (ϵ, δ) -DP if:

$$\Pr[M(D) \in S] \leq e^{\epsilon} \Pr[M(D') \in S] + \delta. \quad (3)$$

In client-level DP-FL, each client computes a local update $\Delta\theta_k^{(t)} = \theta_k^{(t+1)} - \theta_k^{(t)}$ and applies L_2 -norm clipping to bound its sensitivity:

$$\bar{\Delta}\theta_k^{(t)} = \frac{\Delta\theta_k^{(t)}}{\max\left(1, \frac{\|\Delta\theta_k^{(t)}\|_2}{C_t}\right)}. \quad (4)$$

Gaussian noise is then injected before aggregation:

$$\tilde{\Delta}\theta_k^{(t)} = \bar{\Delta}\theta_k^{(t)} + N\left(0, (\sigma C_t)^2 I\right), \quad (5)$$

where σ is the noise multiplier determined by the privacy accountant for a target (ϵ, δ) , and the corresponding noise standard deviation is σC . In many DP-FL methods, the clipping threshold C and noise multiplier σ are set as fixed hyperparameters throughout training. However, under Non-IID heterogeneity, update magnitudes can vary substantially across clients and training phases, making a single static threshold difficult to accommodate. Similarly, in common Non-IID partitioning schemes for federated image classification (e.g., Dirichlet-based partitioning), class proportions across clients are often highly imbalanced, resulting in skewed label distributions. Such statistical heterogeneity is reflected in the optimization process, leading to pronounced scale discrepancies in update norms across clients and communication rounds, and exhibiting stage-wise variations throughout training.

To characterize the effective update strength under fixed configurations, the post-clipping update norm can be written as.

$$\|\bar{\Delta}\theta_k^{(t)}\|_2 = \min\left(\|\Delta\theta_k^{(t)}\|_2, C\right). \quad (6)$$

Based on this, we introduce an intuitive signal-to-noise ratio (SNR) proxy, where the injected Gaussian noise scale is proportional to the clipping bound C under the standard DP-SGD mechanism [26]:

$$SNR_k^{(t)} = \frac{\|\bar{\Delta}\theta_k^{(t)}\|_2}{\sigma C}. \quad (7)$$

Note that Equation (7) is only used as an intuitive proxy to explain the mismatch between effective update magnitude and injected noise; privacy protection is quantitatively evaluated by the MIA ROC-AUC of membership inference attacks in Section 5.

When a portion of clients exhibit update magnitudes below the clipping threshold in most rounds, clipping has limited impact on their updates. However, noise intensity remains governed by the predefined privacy configuration, resulting in low effective update strength. This makes critical information more susceptible to being drowned out by noise, thereby diminishing their training contribution. Conversely, for clients frequently reaching the clipping threshold, their updates typically remain near the threshold scale after clipping. Under identical noise intensity, they exhibit higher effective update strength, granting them greater influence in aggregation. Simultaneously, the distinctive features of their uploaded updates become clearer, making privacy attacks like membership inference easier to execute by providing exploitable discriminative evidence. As training progresses into later stages, the magnitude of updates from most clients generally decreases. Maintaining a fixed threshold configuration will further exacerbate the problem of insufficient effective updates from the majority of clients, slowing convergence and limiting final performance. Simultaneously, the few clients that still maintain large update magnitudes exhibit stronger relative effective update strength, leading to greater update distinguishability and increased privacy attack risks.

4. Methods

4.1. Overall Framework and Design Approach

To mitigate the mismatch between fixed clipping thresholds and static noise calibration under Non-IID heterogeneity—where update magnitudes vary across clients and training stages and a single global configuration can lead to over-clipping or redundant perturbation—we propose AdaCT-DPFL, a differentially private federated learning method with round-wise adaptive clipping thresholds. The overall workflow is illustrated in Figure 1, and can be summarized as a round-wise communication procedure:

- (1) Broadcast: The server broadcasts the current global model $\theta^{(t)}$ to selected clients S_t .
- (2) Local training: Each client $k \in S_t$ performs local training and computes its update $\Delta\theta_k^{(t)}$.
- (3) Norm reporting: Each client computes the update norm $r_k^{(t)} = \|\Delta\theta_k^{(t)}\|_2$ and sends $r_k^{(t)}$ (a scalar) to the server for threshold estimation.
- (4) Threshold estimation: The server aggregates $\{r_k^{(t)}\}_{k \in S_t}$ and computes a round-wise global clipping threshold C_t using a quantile strategy.
- (5) Threshold broadcast: The server broadcasts C_t to clients in S_t .
- (6) Clipping and noise addition: Each client clips its update using C_t and adds Gaussian noise calibrated as $N(0, (\sigma C_t)^2 I)$.
- (7) Aggregation: Clients upload perturbed updates $\bar{\Delta}\theta_k^{(t)}$ to the server, which aggregates them to update the global model θ^{t+1} .

The AdaCT-DPFL algorithm comprises two functional modules:

- (1) Threshold-Adaptive Estimation Module: The server aggregates statistics of client update norms, computes a per-round global clipping threshold using a quantile-based strategy, and broadcasts the threshold to all clients.
- (2) Adaptive Noise Perturbation Module: Each client clips its update using C_t and injects Gaussian noise with a standard deviation of σC_t (where σ is the noise multiplier) before uploading the perturbed update to the server for global aggregation.

To improve clarity and reproducibility, we summarize the complete procedure of AdaCT-DPFL in Algorithm 1.

Algorithm 1. AdaCT-DPFL

Input: Global rounds T ; local epochs E ; participating client set S_t at round t ; quantile coefficient τ , $\tau \in (0, 1)$; noise multiplier σ ; learning rate η ; client datasets $\{D_k\}_{k=1}^K$; initial global model θ^0 .

Output: final global model θ^T .

Server-side

Initialize θ^0

for $t = 0, 1, \dots, T - 1$ **do**

 select participating clients S_t

 broadcast θ^t to all $k \in S_t$

 Phase A (norm reporting): **parallel for** each $k \in S_t$ **do**

$\theta_k^{t+1} \leftarrow \text{Local-Train}(\theta^t, D_k, \eta, E)$

$\Delta\theta_k^{(t)} \leftarrow \text{client model update (Equation (8))}$

$r_k^t \leftarrow \text{norm of the client update (a scalar, Equation (9))}$

 Send scalar r_k^t to server.

end parallel

 form R_t (Equation (10)) set of client update norms

 compute C_t from quantile-based threshold estimation (Equation (11))

 broadcast C_t to all $k \in S$

 Phase B (clipping and perturbation): **parallel for** each $k \in S_t$ **do**

$\bar{\Delta}\theta_k^t \leftarrow \Delta\theta_k^t / \max(1, \|\Delta\theta_k^t\|_2 / C_t)$ (Equation (4))

$\tilde{\Delta}\theta_k^t \leftarrow \bar{\Delta}\theta_k^t + N(0, (\sigma C_t)^2 I)$ (Equation (5))

 Upload $\tilde{\Delta}\theta_k^t$ to server.

end parallel

 aggregate and update

$$\theta^{t+1} \leftarrow \theta^t + \frac{1}{|S_t|} \sum_{k \in S_t} \tilde{\Delta}\theta_k^t$$

end for

Return θ^T .

Note: σ is the noise multiplier; noise standard deviation at round t is σC_t .

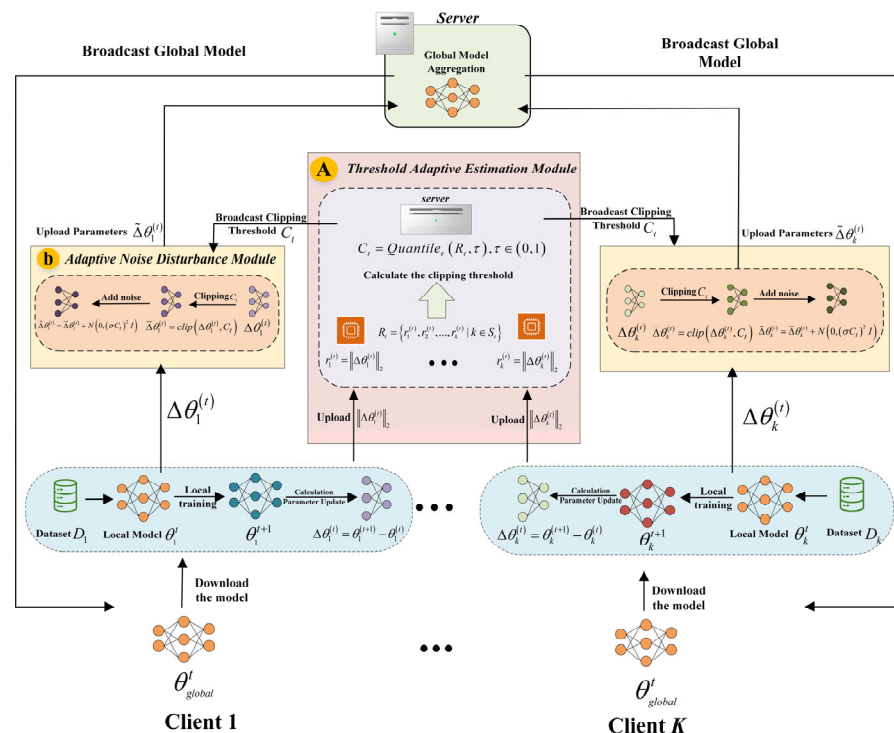


Figure 1. Schematic of the adaptive clipping differentially private federated learning framework.

Next, we detail the two key components in Algorithm 1: (i) quantile-based estimation of the round-wise global clipping threshold C_t (Section 4.2), and (ii) synchronized noise calibration and perturbation of clipped updates (Section 4.3).

4.2. Adaptive Clipping Threshold

Let the set of selected (online) clients at round t be S_t . After receiving the global model θ^t , each client $k \in S_t$ performs local training and obtains a local model $\theta_k^{(t+1)}$. The client update is defined as

$$\Delta\theta_k^{(t)} = \theta_k^{(t+1)} - \theta_k^{(t)}. \quad (8)$$

Each client computes the update norm

$$r_k^{(t)} = \|\Delta\theta_k^{(t)}\|_2. \quad (9)$$

The server collects the norms from participating clients and forms the norm set

$$R_t = \{r_1^{(t)}, r_2^{(t)}, \dots, r_k^{(t)} \mid k \in S_t\}. \quad (10)$$

The round-wise global clipping threshold is then estimated by a quantile strategy:

$$C_t = \text{Quantile}_\tau(R_t, \tau), \quad (11)$$

Here τ controls the clipping intensity. Because C_t is derived from the distribution of client update norms in the current round, it adapts to cross-client heterogeneity and stage-wise training dynamics. When most updates are large (e.g., early training), C_t increases to avoid widespread over-clipping; as training progresses and updates become smaller, C_t decreases accordingly to reduce unnecessary perturbation. Quantile estimation is also robust to a small number of extreme norms, helping prevent abrupt oscillations.

In our design, C_t bounds sensitivity at round t . Noise is calibrated proportionally to C_t (Section 4.3), keeping the effective noise-to-clipping ratio (noise multiplier) consistent for privacy accounting.

4.3. Adaptive Noise Addition

After receiving C_t , each client performs L_2 -norm clipping as shown in Equation (4):

$$\bar{\Delta}\theta_k^{(t)} = \frac{\Delta\theta_k^{(t)}}{\max\left(1, \frac{\|\Delta\theta_k^{(t)}\|_2}{C_t}\right)},$$

This ensures that the clipped update satisfies $\|\bar{\Delta}\theta_k^{(t)}\|_2 \leq C_t$. The clipping operation explicitly bounds the magnitude of each client's uploaded update within C_t , preventing a single client's large update from dominating the global model during aggregation.

By compressing update magnitudes from different clients and training phases into a comparable dynamic range, cross-client update discrepancies are effectively suppressed, thereby weakening the differential statistical cues exploitable by membership inference attacks and improving aggregation stability and convergence performance under Non-IID settings. Moreover, during training, as gradients diminish, the relative impact of injected noise becomes increasingly pronounced; as the number of training iterations increases, the update norms typically decrease accordingly. Therefore, to reduce noise interference for each client model and improve model accuracy, the noise scale is dynamically adjusted

according to the clipping threshold. The noisy update uploaded to the server is defined shown in Equation (5):

$$\tilde{\Delta\theta}_k^{(t)} = \bar{\Delta\theta}_k^{(t)} + N(0, (\sigma C_t)^2 I).$$

where σ denotes the noise multiplier determined by the privacy accountant for a target (ϵ, δ) . Therefore, the Gaussian noise standard deviation at round t is σC_t . This proportional calibration enables the clipping bound and perturbation strength to evolve together across rounds while following a consistent privacy accounting configuration.

Privacy accounting clarification. Although C_t varies across rounds, Equation (5) calibrates noise proportionally to C_t , so the effective noise multiplier used by the accountant remains the same. Consequently, the privacy budget ϵ is accumulated across rounds using the standard RDP accountant configuration (sampling rate, noise multiplier, and number of rounds), while the adaptive threshold mainly improves utility by reducing over-clipping in large-update phases and redundant perturbation in small-update phases.

5. Experiments and Discussion

5.1. Experimental Setup

To validate the effectiveness of the proposed AdaCT-DPFL algorithm, we compare it with two representative client-level differentially private federated learning methods: DP-FedAvg [11] and Fed-DPA [23]. Privacy budgets are uniformly computed and controlled using the Rényi Differential Privacy (RDP) accountant implemented in Opacus 1.5.3. All experiments are conducted in Python 3.10 and PyTorch 2.6.0 on a single NVIDIA GeForce GTX 1660 GPU (NVIDIA Corporation, Santa Clara, CA, USA).

AdaCT-DPFL introduces minimal additional overhead: each client only reports a scalar update norm per round for the global clipping threshold C_t estimation, and the server-side quantile computation for C_t is lightweight over $|S_t|$ scalars. Thus, the method is practical for cross-silo deployments such as multi-section railway condition monitoring, where client participation is limited and communication is relatively stable. In practice, the norm-reporting step can be protected by secure aggregation or by adding small noise to the reported norms, following common practice in adaptive clipping literature. Overall, AdaCT-DPFL does not require changes to model architectures or client data pipelines, making it feasible for real-world deployment.

5.1.1. Attack Model

After federated training, we assume that the global model is deployed as an online inference service. The adversary has black-box query access to the target model: for a given input sample, it can only obtain the predicted probability vector, but has no access to model parameters, gradients, or the training data. In addition, the adversary cannot control any client. The goal is to determine whether a queried sample was included in the target model's training set, i.e., to perform a binary classification between member (participated in training) and non-member (did not participate) samples, thereby conducting a membership inference attack.

Under this threat model, we construct the membership inference dataset for each target model as follows. Samples used during federated training are collected as the member set, while samples from the test set that were not used for training are collected as the non-member set. The target model then performs forward inference on both sets, and we record the output probability vector for each sample. We assign label 1 to member samples and label 0 to non-member samples, forming an attack dataset with member/non-member labels. Finally, the dataset is split into the attacker's training and testing sets to train and evaluate the membership inference attacker.

To exploit privacy-related signals in the target model outputs, we adopt three attack classifiers—Random Forest (RF), Gradient Boosting (GB), and Decision Tree (DT)—using the full predicted probability vector as the input feature. These models learn a mapping from probability distribution patterns to member/non-member labels. Compared with threshold-based attacks that rely on a single statistic (e.g., loss or maximum confidence), using the full probability vector leverages not only the confidence on the true label but also the relative relationships among class probabilities. This enables more fine-grained characterization of overfitting behavior and provides more discriminative evaluation of the privacy protection capability of different DP-FL algorithms under the same experimental setting.

5.1.2. Evaluation Metrics

This paper evaluates the proposed method from two dimensions: model utility and privacy protection. For utility, classification accuracy (Accuracy, denoted as Acc) on the global test set is used to measure the target model's predictive performance. For privacy, the ROC-AUC of the membership inference attacker on the attack test set serves as the attack effectiveness metric, measuring the attacker's ability to distinguish between member and non-member samples, thereby reflecting the target model's privacy leakage risk.

For the global classification model obtained through federated learning, evaluation is conducted using the classification accuracy on the global test set, expressed as:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

where TP , TN , FP , and FN represent true positive, false negative, false positive, and true negative classes, respectively. Accuracy ranges from $[0, 1]$, with higher values indicating better overall prediction performance.

Treating the membership inference attacker as a binary classifier distinguishing “member/non-member,” the area under the ROC curve (ROC-AUC) is used to measure its discriminative capability. The ROC curve plots the false positive rate (FPR) on the x -axis and the true positive rate (TPR) on the y -axis, where

$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}. \quad (13)$$

ROC-AUC provides an overall metric for the $TPR - FPR$ relationship across different decision thresholds, exhibiting threshold-independent characteristics. The ROC-AUC ranges from $[0, 1]$. A higher ROC-AUC value closer to 1 indicates stronger attacker discrimination capability and easier attack success, corresponding to higher privacy leakage risk for the target model. Conversely, a ROC-AUC closer to 0.5 suggests attacker discrimination capability random guessing, indicating better privacy protection effectiveness for the target model.

5.2. Experimental Validation

Two datasets were used for validation.

- (1) **CIFAR-10 Image Dataset:** The CIFAR-10 dataset [27] consists of 10 classes of RGB color images. It contains 60,000 32×32 color images, with 50,000 for training and 10,000 for testing. The ten categories are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck, as shown in Figure 2.

The CIFAR-10 training set is partitioned across clients into Non-IID subsets using a Dirichlet distribution with concentration parameter $\alpha = 1$, serving as each client's local training data. The CIFAR-10 test set is used to evaluate the global model performance. On

CIFAR-10, we employ a convolutional neural network consisting of three convolutional layers and three fully connected layers. A lightweight CNN is employed for CIFAR-10 experiments to ensure training stability under DP noise and fair performance comparisons among DP-FedAvg, Fed-DPA and AdaCT-DPFL. Importantly, AdaCT-DPFL modifies the clipping-threshold estimation and noise calibration mechanism rather than the model architecture. Therefore, the proposed method is architecture-agnostic and can be integrated with stronger backbones (e.g., ResNet-style networks) and realistic FL workloads without changing its core design. We will further evaluate larger backbones and cross-device settings in future work.

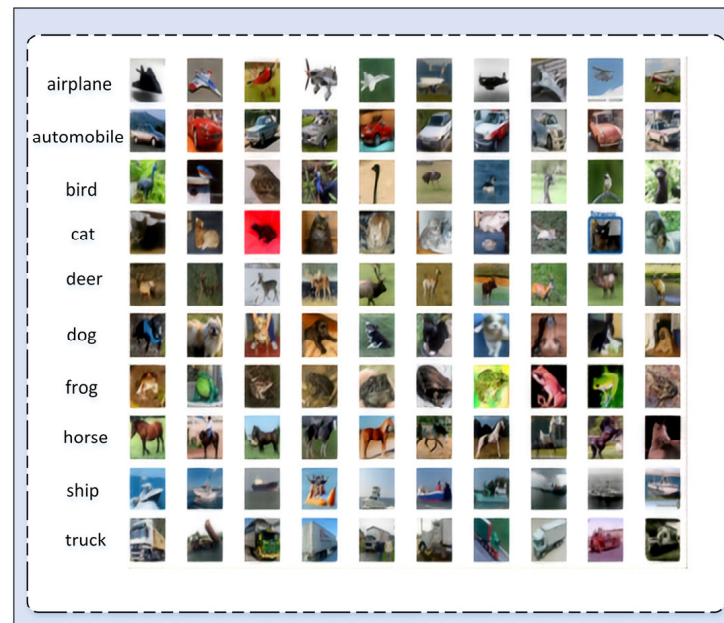


Figure 2. CIFAR-10 Dataset Examples.

(2) Real-world rail defect dataset

The experimental dataset consists of rail vibration signals collected from multiple track sections managed by a Chinese railway locomotive depot. Accelerometers mounted on a rail defect inspection vehicle recorded vibration responses induced by wheel–rail interactions during train operation. Each track section is treated as one client, and its data form the local dataset of that client, thereby constructing a cross-section federated learning setting. Sample labels are determined based on on-site maintenance records and manual verification, covering three typical defect types: rail corrugation, fish-scale defects, and spalling. The raw vibration signals are first processed by outlier removal and normalization, and then segmented into fixed-length time windows to construct input samples. The data acquisition and preprocessing pipeline is illustrated in Figure 3.

After client partitioning, each client further splits its data into training and testing sets with an 8:2 ratio. The training samples remain on the client for local training, while the testing samples are aggregated to form a global test set for unified evaluation. Due to variations in inspection frequency, operating conditions, and sensor installation across different track sections, clients exhibit both sample-size imbalance and feature distribution shifts, resulting in a typical non-IID setting. On the rail defect dataset, we employ a lightweight convolutional neural network consisting of two convolutional layers and two fully connected layers.

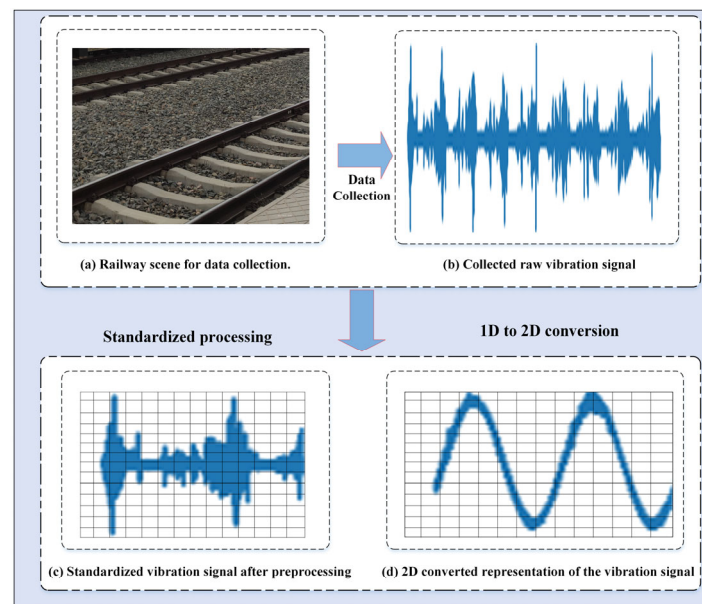


Figure 3. Example of the Rail Dataset.

5.2.1. Analysis of CIFAR-10 Dataset Experimental Results

Experimental parameters for the CIFAR-10 dataset were set as follows: 10 clients, 30 communication rounds, 4 local training epochs per round, a batch size of 64, and a learning rate of 1×10^{-3} .

Membership inference attacks were conducted on the target models trained by DP-FedAvg, Fed-DPA, and the proposed AdaCT-DPFL method under different privacy budgets ($\epsilon = 4, 8, 12, 16$). Results are reported in Table 1. As ϵ increases, the ROC-AUC of membership inference attacks generally increases, indicating that attacks become more successful under weaker privacy protection. Under the same privacy budget, the proposed method consistently achieves lower ROC-AUC across different attacker configurations and remains closer to 0.5 overall, suggesting that the attacks approach random guessing and that the proposed method provides stronger protection against membership inference. This improvement is attributed to adaptive clipping, which suppresses the stage-wise dominance of a small number of anomalous updates during aggregation in non-IID settings, thereby weakening memorization of training samples and reducing discriminative cues exploitable by attackers.

Table 1. ROC-AUC Comparison Results for Membership Inference Attacks on the CIFAR-10 Dataset.

Attack Model	Method	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 12$	$\epsilon = 16$	Average
RF	DP-FedAvg	0.539	0.548	0.555	0.559	0.549
	Fed-DPA	0.541	0.549	0.553	0.556	0.550
	AdaCT-DPFL	0.525	0.529	0.535	0.541	0.533
GB	DP-FedAvg	0.545	0.552	0.561	0.569	0.557
	Fed-DPA	0.543	0.547	0.554	0.558	0.551
	AdaCT-DPFL	0.528	0.536	0.540	0.548	0.538
DT	DP-FedAvg	0.530	0.538	0.544	0.548	0.540
	Fed-DPA	0.525	0.530	0.533	0.538	0.532
	AdaCT-DPFL	0.515	0.521	0.526	0.532	0.524

Table 2 reports the classification accuracy of the three algorithms on the CIFAR-10 dataset under different privacy budgets. As ϵ increases, the accuracy of all three methods improves overall, indicating that the target model attains better utility under weaker

privacy protection (i.e., reduced relative noise perturbation). Compared with DP-FedAvg and Fed-DPA, the proposed method achieves higher classification accuracy across all privacy budget settings. This gain mainly arises from the dynamic matching of the adaptive clipping threshold and adaptive noise scale across training stages: it mitigates information loss caused by over-clipping in the early stage, while reducing the interference of injected noise on small-magnitude updates in the later stage. Consequently, the proposed method improves classification accuracy under the same privacy budget.

Table 2. Accuracy Comparison of Three Algorithms on the CIFAR-10 Dataset.

Method	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 12$	$\epsilon = 16$
DP-FedAvg	51.45	53.49	54.26	55.31
Fed-DPA	54.34	57.42	58.18	58.35
AdaCT-DPFL	55.18 $\uparrow$ 0.84	57.92 $\uparrow$ 0.5	58.60 $\uparrow$ 0.42	58.89 $\uparrow$ 0.54

Note: The upward arrows ($\uparrow$) with numerical values indicate the accuracy improvement of the proposed AdaCT-DPFL method compared to the baseline method under the same privacy budget ϵ .

Figure 4 shows the accuracy convergence curves over communication rounds for the three algorithms on the CIFAR-10 dataset. Compared with DP-FedAvg and Fed-DPA, the proposed method achieves a higher and more stable final accuracy. These results suggest that adaptive thresholding stabilizes the training dynamics and enables more efficient convergence under non-IID settings.

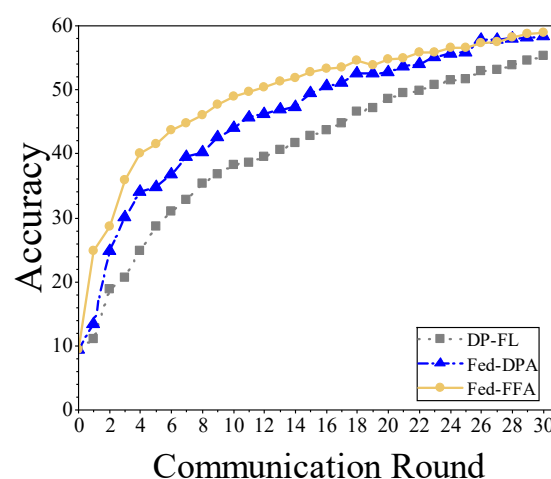


Figure 4. Comparison of accuracy convergence curves on the CIFAR-10 dataset.

In summary, the results on CIFAR-10 demonstrate that, under the same privacy budget, the proposed method further reduces the discriminative power of membership inference attacks while effectively improving classification accuracy. Moreover, it exhibits more stable convergence during training, thereby validating the effectiveness of the proposed approach.

5.2.2. Analysis of Experimental Results on the Rail Dataset

For the rail defect dataset experiments, the hyperparameters were set as follows: 10 clients, 40 rounds of global communication, 5 local training epochs per round, a batch size of 16, and a learning rate of 1×10^{-3} .

Membership inference attacks were conducted on target models trained by DP-FedAvg, Fed-DPA, and the proposed AdaCT-DPFL under different privacy budgets ($\epsilon = 4, 8, 12, 16$), with results reported in Table 3. Under the same ϵ , AdaCT-DPFL achieved lower ROC-AUC across all three attackers (RF, GB, and DT). Averaged over all ϵ settings, AdaCT-DPFL

achieved an average ROC-AUC of 0.528, which is lower than that of DP-FedAvg and Fed-DPA by 0.9% and 0.4%, respectively, indicating that the attackers' discrimination capability becomes closer to random guessing. These results demonstrate the stronger privacy protection of the proposed method.

Table 3. ROC-AUC Comparison of Membership Inference Attacks on the Rail defect Dataset.

Attack Model	Method	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 12$	$\epsilon = 16$	Average
RF	DP-FedAvg	0.522	0.531	0.548	0.558	0.540
	Fed-DPA	0.521	0.526	0.537	0.545	0.532
	AdaCT-DPFL	0.519	0.523	0.533	0.539	0.529
GB	DP-FedAvg	0.529	0.536	0.550	0.562	0.544
	Fed-DPA	0.525	0.529	0.544	0.553	0.538
	AdaCT-DPFL	0.524	0.531	0.541	0.548	0.536
DT	DP-FedAvg	0.516	0.520	0.531	0.538	0.526
	Fed-DPA	0.517	0.521	0.528	0.538	0.526
	AdaCT-DPFL	0.509	0.517	0.524	0.533	0.521

Table 4 presents the classification accuracy comparison results of the three algorithms on the rail defect dataset. Under different ϵ settings, the proposed method consistently achieves higher accuracy than DP-FedAvg and Fed-DPA. Taking $\epsilon = 4$ as an example, the proposed method achieves an accuracy of 85.86%, representing improvements of 4.16% and 1.34% over DP-FedAvg and Fed-DPA, respectively. Under a larger privacy budget ($\epsilon = 16$), the proposed method achieves an accuracy of 88.84%, representing improvements of 2.38% and 1.04% over DP-FedAvg and Fed-DPA, respectively.

Table 4. Accuracy Comparison of Three Algorithms on the Rail Stripping Damage Dataset.

Method	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 12$	$\epsilon = 16$
DP-FedAvg	81.70	83.63	84.52	86.46
Fed-DPA	84.52	84.59	86.61	87.80
AdaCT-DPFL	85.86 $\uparrow$ 1.34	86.46 $\uparrow$ 1.87	88.54 $\uparrow$ 1.93	88.84 $\uparrow$ 1.04

Note: The upward arrows ($\uparrow$) with numerical values indicate the accuracy improvement of the proposed AdaCT-DPFL method compared to the baseline method under the same privacy budget ϵ .

Figure 5 presents the accuracy convergence curves on the rail defect dataset. Compared with DP-FedAvg and Fed-DPA, the proposed method captures informative patterns more effectively in the early training phase and reaches higher accuracy within the same number of communication rounds. During the middle and late stages, DP-FedAvg and Fed-DPA exhibit less stable convergence with more pronounced fluctuations. In contrast, AdaCT-DPFL achieves faster and more stable convergence by coordinating adaptive clipping thresholds and adaptive noise scaling to better accommodate heterogeneous update magnitudes across clients, thereby reducing aggregation oscillations caused by mismatched clipping and redundant noise.

In summary, this section compares DP-FedAvg, Fed-DPA, and the proposed method on the rail defect dataset from three aspects: resistance to membership inference attacks, classification accuracy, and convergence behavior. The results show that, under the same privacy budget, the proposed method further reduces the attacker's discriminative capability and achieves better accuracy and more stable convergence than the baselines, demonstrating superior overall performance under non-IID settings with pronounced client heterogeneity.

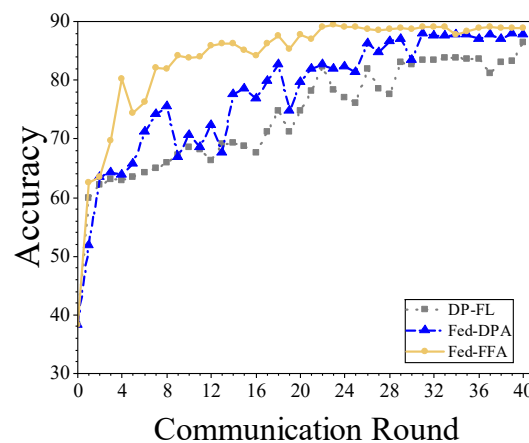


Figure 5. Accuracy convergence curves comparison on the Rail Stripping Damage Dataset.

6. Conclusions

This paper proposes AdaCT-DPFL, a differentially private federated learning framework with quantile-based adaptive clipping thresholds and synchronized noise scaling. By dynamically estimating the clipping threshold according to the distribution of client update norms in each communication round, the proposed method mitigates the mismatch between heterogeneous update magnitudes and fixed privacy configurations under Non-IID settings. Extensive experiments on CIFAR-10 and a real-world rail defect dataset demonstrate that, under identical privacy budgets, AdaCT-DPFL achieves improved privacy–utility trade-offs compared with DP-FedAvg and Fed-DPA. Specifically, it reduces the discriminative capability of membership inference attacks while maintaining competitive and more stable classification performance. The adaptive threshold mechanism effectively alleviates over-clipping in early training stages and redundant noise perturbation in later stages. However, this work has several limitations. First, the current privacy analysis is primarily based on RDP accounting under a round-wise adaptive threshold, and a tighter theoretical bound for dynamically varying sensitivity remains to be investigated. Second, experiments are conducted under moderate-scale FL settings; larger cross-device scenarios with partial participation should be further explored. Third, stronger attack models, including adaptive adversaries and gradient-based inference attacks, should be evaluated. Future research will focus on (1) deriving formal privacy bounds under adaptive sensitivity; (2) extending the framework to personalized and hierarchical FL architectures; (3) integrating secure aggregation to provide end-to-end protection; and (4) exploring adaptive threshold learning mechanisms beyond quantile estimation, such as learning-based or Bayesian threshold selection strategies.

Author Contributions: Conceptualization, Y.Z., J.L. and Z.W.; data collection and analysis, Y.Z., Y.T. and W.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, J.L. and Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (No. 52272347) and the National Natural Science Foundation of China of Hunan Province (No. 2024JJ7132).

Data Availability Statement: The datasets used in this study are publicly available. The CIFAR-10 dataset can be accessed at <https://www.kaggle.com/datasets/pankrzysiu/cifar10-python> (accessed on 11 March 2026).

Conflicts of Interest: The authors declare that this study was conducted without any commercial or financial relationships that could be considered potential conflicts of interest.

References

- McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*; PMLR: New York, NY, USA, 2017.
- Wen, J.; Zhang, Z.; Lan, Y.; Cui, Z.; Cai, J.; Zhang, W. A survey on federated learning: Challenges and applications. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 513–535. [[CrossRef](#)] [[PubMed](#)]
- Liu, B.; Lv, N.; Guo, Y.; Li, Y. Recent advances on federated learning: A systematic survey. *Neurocomputing* **2024**, *597*, 128019. [[CrossRef](#)]
- Zhang, R.; Guo, S.; Wang, J.; Xie, X.; Tao, D. A survey on gradient inversion: Attacks, defenses and future directions. *arXiv* **2022**, arXiv:2206.07284. [[CrossRef](#)]
- Carlini, N.; Chien, S.; Nasr, M.; Song, S.; Terzis, A.; Tramèr, F. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*; IEEE: New York, NY, USA, 2022.
- Zhao, J.; Bagchi, S.; Avestimehr, S.; Chan, K.; Chaterji, S.; Dimitriadis, D.; Li, J.; Li, N.; Nourian, A.; Roth, H. The federation strikes back: A survey of federated learning privacy attacks, defenses, applications, and policy landscape. *ACM Comput. Surv.* **2025**, *57*, 230. [[CrossRef](#)]
- Rodríguez-Barroso, N.; Jiménez-López, D.; Luzón, M.V.; Herrera, F.; Martínez-Cámara, E. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Inf. Fusion* **2023**, *90*, 148–173. [[CrossRef](#)]
- Nguyen, D.H.; Nguyen, P.L.; Nguyen, T.T.; Pham, H.H.; Tran, D.A. FedBlock: A Blockchain Approach to Federated Learning against Backdoor Attacks. In *2024 IEEE International Conference on Big Data (BigData)*; IEEE: New York, NY, USA, 2024; pp. 7981–7990.
- Bai, L.; Hu, H.; Ye, Q.; Li, H.; Wang, L.; Xu, J. Membership inference attacks and defenses in federated learning: A survey. *ACM Comput. Surv.* **2024**, *57*, 89. [[CrossRef](#)]
- Ren, X.; Yang, S.; Zhao, C.; McCann, J.; Xu, Z. Belt and braces: When federated learning meets differential privacy. *Commun. ACM* **2024**, *67*, 66–77. [[CrossRef](#)]
- Geyer, R.C.; Tassilo, K.; Nabi, M. Differentially private federated learning: A client level perspective. *arXiv* **2017**, arXiv:1712.07557.
- El Ouadrhiri, A.; Abdelhadi, A. Differential privacy for deep and federated learning: A survey. *IEEE Access* **2022**, *10*, 22359–22380. [[CrossRef](#)]
- Ilya, M. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*; IEEE: New York, NY, USA, 2017; pp. 263–275.
- Ozturk, O.; Buyuktanir, B.; Baydogmus, G.K.; Yildiz, K. Differential Privacy in Federated Learning: Mitigating Inference Attacks with Randomized Response. *arXiv* **2025**, arXiv:2509.13987. [[CrossRef](#)]
- Li, Y.; Chang, T.H.; Chi, C.Y. Secure federated averaging algorithm with differential privacy. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*; IEEE: New York, NY, USA, 2020.
- Ponomareva, N.; Hazimeh, H.; Kurakin, A.; Xu, Z.; Denison, C.; McMahan, H.B.; Vassilvitskii, S.; Chien, S.; Thakurta, A.G. How to dp-fy ml: A practical guide to machine learning with differential privacy. *J. Artif. Intell. Res.* **2023**, *77*, 1113–1201. [[CrossRef](#)]
- Kucur, E.N.; Buyuktanir, T.; Ugurelli, M.; Yildiz, K. Privacy-Preserving Machine Learning Techniques: Cryptographic Approaches, Challenges, and Future Directions. *Appl. Sci.* **2025**, *16*, 277. [[CrossRef](#)]
- Yuan, J.; Chen, Y.; Wang, Z.; Wang, C.; Hu, X.; Zeng, Z. DP-FedPUAC: Federated learning with differential privacy via adaptive gradient clipping and local iteration optimization. *Inf. Sci.* **2025**, *733*, 122981. [[CrossRef](#)]
- Yue, G.; Yan, L.; Kang, L.; Shen, C. AdapLDP-FL: An Adaptive Local Differential Privacy for Federated Learning. *IEEE Trans. Mob. Comput.* **2025**, *24*, 5569–5583. [[CrossRef](#)]
- Zhou, H.; Zheng, Y.; Jia, X. Towards robust and privacy-preserving federated learning in edge computing. *Comput. Netw.* **2024**, *243*, 110321. [[CrossRef](#)]
- Huang, X.; Ding, Y.; Jiang, Z.L.; Qi, S.; Wang, X.; Liao, Q. DP-FL: A novel differentially private federated learning framework for unbalanced data. *World Wide Web* **2020**, *23*, 2529–2545. [[CrossRef](#)]
- Wu, X.; Zhang, Y.; Shi, M.; Li, P.; Li, R.; Xiong, N.N. An adaptive federated learning scheme with differential privacy preservation. *Future Gener. Comput. Syst.* **2022**, *127*, 362–372. [[CrossRef](#)]
- Yang, X.; Huang, W.; Ye, M. Dynamic personalized federated learning with adaptive differential privacy. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 72181–72192.
- Li, T.; Zaheer, M.; Liu, K.Z.; Reddi, S.J.; McMahan, H.B.; Smith, V. Differentially private adaptive optimization with delayed preconditioners. *arXiv* **2022**, arXiv:2212.00309.
- Andrew, G.; Thakkar, O.; McMahan, B.; Ramaswamy, S. Differentially private learning with adaptive clipping. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 17455–17466.

26. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*; IEEE: New York, NY, USA, 2016.
27. Krizhevsky, A. *Learning Multiple Layers of Features From Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.