Journal of Information Security and Applications 70 (2022) 103309

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

High-accuracy low-cost privacy-preserving federated learning in IoT systems via adaptive perturbation

Tian Liu [a,b], Xueyang Hu [a], Hairuo Xu [a], Tao Shu [a,*], Diep N. Nguyen [c]

[a] Department of Computer Science and Software Engineering, Auburn University, Auburn, AL, USA
[b] Zhejiang Laboratory, Hangzhou, China
[c] School of Electrical and Data Engineering, University of Technology Sydney, NSW, Australia

ARTICLE INFO

Keywords:
Federated learning
Privacy-preserving
IoT
Convergence performance
Information leakage
Local privacy

ABSTRACT

With the rapid development of the Internet of Things (IoT), federated learning (FL) has been widely used to obtain insights from collected data while preserving data privacy. Differential privacy (DP) is an additive noise scheme that has been widely studied as a privacy-preserving approach on FL. However, privacy protection under DP usually comes at the cost of model accuracy for the underlying FL process. In this paper, we propose a novel low-cost (for both communication and computational overhead) adaptive noise perturbation/masking scheme to protect FL clients' privacy without degrading the global model accuracy. In particular, we set the magnitude of the additive noise to adaptively change with the magnitude of the local model updates. Then, a direction-based filtering scheme is used to accelerate the convergence of the FL model. A maximum tolerable noise bound for local clients is derived using the central limit theorem (CLT). The designed noise maximizes privacy protection for clients while preserving the accuracy and convergence rate of the FL model , as a result of the noise cancelling out and forming a more concentrated distribution after the aggregation operation on the server. We theoretically prove that FL with the proposed noise perturbation scheme retains the same accuracy and convergence rate ($\mathcal{O}(1/T)$ for convex loss functions and $\mathcal{O}(1/\sqrt{T})$ for non-convex loss functions) as that of non-private FL with SGD. We also evaluate the performance of the proposed scheme in terms of convergence behavior, computational efficiency, and privacy protection against state-of-the-art privacy inference attacks on real-world datasets. Experimental results show that FL with our proposed perturbation scheme outperforms DP in the accuracy and convergence rate of the FL model in both client dropout and non-client dropout scenarios. Compared with DP, our proposed scheme does not incur additional computational and communication overhead. Our approach provides DP-comparable or better effectiveness in defending against privacy attacks under the same global model accuracy.

1. Introduction

The development of the Internet of Things (IoT) enables the connection of a wide range of devices to the Internet [1] to provide ubiquitous sensing and computation capabilities. The data collected by these devices can be used to train machine learning models. Although the data on one device may be insufficient to obtain a satisfactory model, the data on other devices can be benefited via network communication. Federated learning (FL) [2,3] allows a machine learning algorithm to learn from data stored on a wide range of physically separated devices. Technically, FL is a distributed learning system, which allows multiple local clients to collaboratively train a high-accuracy global model by taking advantage of a wide range of data without sharing their local collected data. FL has found its applications in most emerging services and systems, e.g., in mobile applications such as next-word and emoji prediction on smartphones [4–6], environmental monitoring [7], smart healthcare [8,9], and smart city [10].

Although clients do not directly reveal their private data, shared model updates can unintentionally leak sensitive information about the data on which they were trained [11]. As pointed out by previous studies, using FL scheme alone is insufficient in protecting the clients' local data privacy. For example, from the FL model, an adversary can infer whether a given data sample was presented in the training data or not [12,13], recover a representative data sample used in the training [14], or infer property information about the client's local training data [15].

* Corresponding author.
E-mail addresses: tianliu@auburn.edu (T. Liu), xueyang.hu@auburn.edu (X. Hu), hairuoxu@auburn.edu (H. Xu), tshu@auburn.edu (T. Shu), Diep.Nguyen@uts.edu.au (D.N. Nguyen).

https://doi.org/10.1016/j.jisa.2022.103309

Available online 1 September 2022

Ideally, FL with a privacy-preserving mechanism on IoT devices, such as smartphones, smart watches, and cameras, should take into account the following **constraints**: (1) computational capacity is limited, so computationally expensive encryption algorithms are unaffordable; (2) devices have limited power supply and network connectivity; (3) clients are flexible to join or leave training, so dropouts are common.

Several studies have focused on how to preserve privacy in FL. But none of them can fully address the aforementioned constraints. In particular, the main approaches are secure multiparty computation (MPC) and differential privacy (DP). A branch in MPC is based on homomorphic encryption. Paillier cryptosystem is an additive homomorphic encryption algorithm [16–18], which naturally matches the aggregation operation in FL. But the main drawback is its high computational complexity. Another approach uses secret sharing [18], which is relatively computationally efficient and can also handle client dropout. However, the requirement of information exchange between each pair of clients makes this approach impractical in moderate to large-scale IoT systems. DP is a promising solution that injects random noise into the data or model updates, providing a statistical privacy guarantee for individual records and privacy protection against inference attacks. However, privacy protection comes at the cost of model accuracy. Additionally, one challenge in training with DP is choosing an appropriate clipping bound. An inappropriate clipping bound can degrade model accuracy or even prevent a model from converging due to the bias introduced by the clipping operation [19].

In this work, we propose a novel low cost (for both communication and computational overhead) adaptive noise-perturbation privacy-preserving scheme, which does not sacrifice FL model accuracy for privacy, while enjoying a DP-comparable or in some cases better privacy protection. More specifically, our scheme protects local privacy by adding random noise to each local model update (i.e., perturbing local model update by adding random noise). These random noises are deliberately designed so that individually they can provide sufficient protection for the privacy of each local model. But when combined at the FL server, the aggregation of these noises will present a cancel-out effect by the central limit theorem (CLT), so that the aggregated noises at the server are more condensed and help to preserve the global model accuracy. In real FL applications, the number of clients is much larger than 30, which is considered sufficient for CLT to hold. In addition, unlike random noise in the DP scheme, our noise masking scheme takes both magnitude and direction into consideration when adding noise to local model updates to retain high global model accuracy and expedite global model convergence. Specifically, we introduce an adaptive noise scaling method that sets the magnitude of the random noise proportional to the magnitude of the local model updates, i.e., the magnitude of noise changes with that of local model updates at the same rate, which ensures sufficient privacy protection while preventing the introduction of excessive noise, especially when the FL model is close to convergence. To maintain the same convergence rate and accuracy as in regular FL, the noise scale is chosen on the basis of the number of participating clients, so that the magnitude of the aggregation of noise does not exceed the magnitude of local model updates. Moreover, we monitor the angular distance, calculated from cosine similarity, between the true local model updates and the noise-perturbed local model updates. Noise with a large angular distance will be filtered out, making it easier for the global model to converge. With deliberately chosen noise magnitude and angular distance, the FL with the proposed noise scheme achieves the same convergence performance as the regular FL and similar or better privacy protection compared to state-of-the-art DP frameworks [20,21].

To the best of our knowledge, we are the first to take both magnitude and direction into consideration aiming at protecting FL clients' privacy while preserving the FL model accuracy. Our **contribution** in this paper is threefold:

- For a strongly convex loss function, we prove that a noise-perturbed FL is guaranteed to converge to the same value as the regular FL (i.e., there is no accuracy loss) as long as the magnitude of the added noise is proportional to the magnitude of the local model update. Given the number of clients participating in the perturbed FL, we also derive the maximum tolerable variance of the added noise at individual clients that guarantees that the magnitude of the aggregated noise at the FL server does not exceed the magnitude of the aggregation of all local model updates (i.e., the direction of descent is still preserved), so that the perturbed FL maintains the same convergence rate $\mathcal{O}(1/T)$ as that of SGD on convex loss functions. These theoretical findings enable us to develop the proposed adaptive noise perturbation scheme that maximizes privacy protection for clients while maintaining the same accuracy as that of regular FL. We also provide a statistical method to select the angular distance threshold based on the dimension of the model updates to accelerate the convergence of the perturbed FL.
- For the non-convex loss function scenario, we derive the worst-case convergence bound for FL under the proposed noise perturbation scheme. This bound shows that the noise-perturbed learning process converges at a rate of $\mathcal{O}(1/\sqrt{T})$, the same as that of an SGD on non-convex functions. With the proposed angular distance filtering scheme, our proof indicates that the actual convergence is faster than the derived worst-case convergence bound.
- Extensive experiments are conducted on MNIST and CIFAR-10 datasets to validate our theoretical convergence analyses and evaluate the time and computational efficiency, as well as the effectiveness of the proposed scheme in defending against state-of-the-art privacy inference attacks. The numerical results show that the proposed scheme outperforms DP in convergence rate and accuracy in both dropout and non-dropout scenarios, which are consistent with our theoretical convergence analyses. The proposed scheme does not incur extra computational and communication overhead compared with DP. Our proposed noise perturbation scheme provides comparable or, in many cases, stronger privacy protection than DP, under the same global model accuracy.

The rest of this paper is organized as follows. Section 2 briefly reviews the FL and related work. Section 3 presents our threat model and security goals. Section 4 describes our proposed additive noise scheme. Theoretical convergence analyses are provided in Section 5. The settings and results of the experiments are presented in Section 6 and Section 7, respectively. We conclude our work and recommend future research directions in Section 8. And detailed proofs of our key findings are given in the Appendix.

Throughout this paper, we use the following **notation**:

- $\|\cdot\|$ denotes the ℓ_2 norm.
- $<_\epsilon$ denotes slightly greater than. $a <_\epsilon b$ means $b = a + \epsilon$, where $\epsilon \in \mathbb{N}^+$.
- D denotes the global data and is distributed to N clients, where $D = \cup_{n=1}^{N} D_n$, and D_n denotes the data on the nth client. A subset of K clients ($K < N$) is selected to participate in a round of FL training.
- $F_k(\cdot)$ and $F(\cdot)$ denote the loss function on the client k and the global loss function, respectively.
- $\nabla F_k(\cdot)$ and $\nabla F(\cdot)$ denote the gradients of the local loss function and the global loss function, respectively.
- $w_k^{T,\tau}$ denotes the local model weight of client k in τ-th local step in Tth global aggregation, and w^T denotes the global model weight in Tth global aggregation.
- $\tilde{w}_k^T$ and $\tilde{w}^T$ denote the noise-perturbed local model weight and the noise-perturbed global model weight at Tth aggregation, respectively.
- r_k^T denotes the additive noise in the client k in the Tth global aggregation.

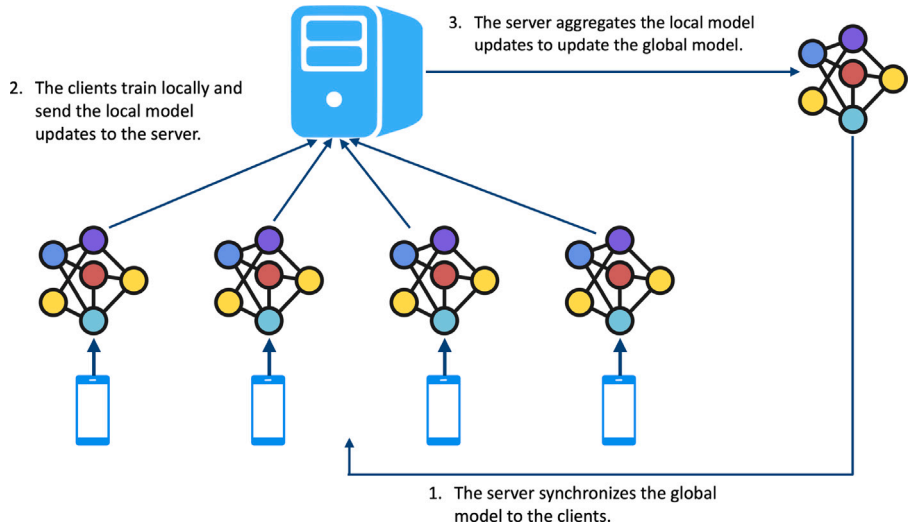


Fig. 1. An illustration of the FL process.

2. Preliminary and related work

2.1. Federated learning

The global data $D = \cup_{n=1}^{N} D_n$ are distributed to N clients and each client maintains its local data D_n. Each time, K ($K \leq N$) of N clients are selected to participate in the training. Specifically, each client maintains a local model trained from the local training dataset. A central server maintains a global model by aggregating local model updates from participating clients in each round. The objective of FL training is to minimize the loss:

$$F(w) = \sum_{k=1}^{K} F_k(w), \tag{1}$$

by optimizing over the model parameter w, where $F_k(w)$ is the loss function on the local data of the kth client:

$$F_k(w) = \frac{1}{|D_k|} \sum_{(x,y)\in D_k} L(w;(x,y)),\ k \in [K], \tag{2}$$

where L is the empirical loss function. Here, we describe FedAvg, which is probably the most widely used FL algorithm. FedAvg iteratively performs the following three steps (illustrated in Fig. 1):

2.1.1. Global model synchronization

In the Tth global aggregation, the central server randomly selects K from N clients and broadcasts the latest global model w^T to selected clients: $w_k^{T,0} \leftarrow w^T$.

2.1.2. Local model training

Each client k updates its own local model w_k by running an SGD on the local dataset D_k for t steps. The τ-th step on the client k follows :

$$w_k^{T,\tau+1} \leftarrow w_k^{T,\tau} - \eta \nabla F_k(w_k^{T,\tau}, \xi_k^{T,\tau}), \tag{3}$$

where $\xi_k^{T,\tau}$ is a mini-batch of samples randomly chosen from the local dataset D_k, and η is the local learning rate.

2.1.3. Global model update

After performing local training for t steps, the client transmits the model updates $\Delta w_k^T = w_k^{T+t} - w^T$ back to the central server. The central server then updates the global model by performing a weighted average on the local model updates sent from K clients :

$$w^{T+1} \leftarrow w^T + \sum_{k=1}^{K} \frac{n_k}{n} \Delta w_k^T, \tag{4}$$

where $n_k = |D_k|$ is the number of training data on the client k and $n = \sum_{k=1}^{K} n_k$ is the total number of training data used on the selected clients.

2.2. Privacy-preserving FL

Existing work in privacy-preserving FL can be classified into two categories: secure multiparty computation and differential privacy.

Secure multiparty computation (MPC). Existing work utilizes homomorphic encryption [17,17,22,23] and secret sharing [16,18,24,25] to preserve privacy in FL. With additive homomorphic encryption, for example, the Paillier cryptosystem, the server can perform gradient aggregation without decrypting them. Before training starts, the HE key pair is distributed to each client through a secure channel. In each training iteration, each client calculates the local model update, encrypts it with the public key, and uploads the ciphertext to the server. The server aggregates the encrypted gradients from all clients and sends the results back to the clients. Each client decrypts the received ciphertext using the private key to obtain global model updates due to additive homomorphism. But such algorithms are computationally expensive. FL systems with homomorphic encryption suffer from extremely high computational overhead and can hardly be applied on IoT devices. Scholars in [26] used secret sharing for secure aggregation in FL, allowing K parties to obtain the output of a function based on their K inputs while preventing any leakage of inputs other than the outputs. In [18], a noninteractive secure aggregation protocol was proposed based on secret sharing and key agreement, but a trusted authority was required. And the researchers in [16] proposed a double masking scheme that supports verification. The weakness of secret sharing lies in the communication cost. Each client needs to send a secret share to the majority of participating clients to guarantee the robustness of the model, or each pair of clients needs to communicate and agree on some random masks. Neither of them is applicable to IoT systems, in which devices hardly have direct communications.

Despite the high computational and communication overhead, such MPC approaches do not eliminate FL information leakage. In FL with homomorphic encryption, the server may collude with clients to decrypt local model updates from the ciphertext. As for secret sharing, the adversary still has a chance to infer the input information from the output of the function since the function usually does not change.

Differential privacy (DP). Differential privacy [23,27] is a noise perturbation mechanism that provides a statistical privacy guarantee for individual records. Existing work incorporates DP into FL from different perspectives. Shokri et al. [28] were the first scholars to integrate differential privacy into deep learning to protect training data privacy. NbAFL was proposed in [29] to protect uplink and downlink communication. In [30], 2DP-FL was proposed to handle non-i.i.d. distributions among clients and could adapt to different privacy needs. Naseri et al. [31] evaluated the feasibility and effectiveness of local and central differential privacy (LDP/CDP) in FL. In [32], a user-level differential privacy (UDP) was proposed to allow adjustable privacy protection for each FL participant. It has been empirically shown in [12] that DP is effective in defending against membership inference attacks [26,33], reconstruction attacks [34], and model inversion attacks [14], but is merely effective in defending against property inference attacks [31].

Additionally, in DP, bounding the influence of a single client is necessary for both privacy and the utility of the model. The choice of the bounding threshold, i.e., the clipping bound, has decisive effects on both privacy and model utility, due to the fact that the clipping bound could introduce bias to model updates [19]. Existing work quantifies the bias in ℓ_∞ [35] and ℓ_2 [36]. Nissim et al. [37] used a calibrated noise according to smooth sensitivity, but requires additional knowledge and communication of the original model updates. Adaptive clipping bounds that utilize the statistics of model updates to track and predict its change were proposed in [21,38], but such clipping bounds do not immediately react to the change in model updates, which could still result in excessive noise injection. Moreover, none of the existing work investigated the impact of the direction of the additive random noise on the convergence of the model.

2.3. Privacy attacks against FL

We mainly discuss two privacy attacks in FL: the membership inference attack and the property inference attack.

Membership inference attack. Shokri et al. [39] demonstrated that an adversary can infer whether or not a given data sample was presented in the training data by the difference in the response of the model. Specifically, a binary classifier, called a shadow model, is trained for each output class using the same machine learning algorithm. Each shadow model identifies the membership of data samples of the corresponding class by outputting the probabilities over the membership and nonmembership classes. Studies in [40–43] demonstrated the leakage of membership in various areas. Studies in [13,26,33,44] analyzed membership inference from the perspectives of generative models, transferability, the relationship with overfitting, and defenses, respectively.

Property inference attack. The property inference attack was first proposed by Ateniese et al. [45] against Hidden Markov Models and Support Vector Machine classifiers. Ganju et al. in [46] designed a property inference attack on fully connected networks. The adversary trains *meta-classifier* to classify *target classifier* depending on whether or not it has the property. In [47,48], a training label composition inference attack was proposed. The adversary could infer the composition of the training label of a client's private data by finding a label composition such that the synthesized model updates are close to the true model update as much as possible.

3. Problem setup

3.1. Threat model

We consider a potential threat of privacy inference attack during the learning process. Specifically, an adversary could infer information about clients' private data through the model information exchange between clients and the server. Our proposed method is designed to withstand two potential adversaries: the central server and eavesdroppers.

- **Honest-but-curious server.** We assume that the central server is honest-but-curious, meaning that the server follows the FL protocol, but may try to infer some private information from the client's model updates.
- **Eavesdroppers.** We also consider the potential attack, in which an eavesdropper monitors the communication link between clients and the server. We assume that the attacker has no access to client's training data, but can eavesdrop model updates from the communication between clients and the server and infer private information about clients.

3.2. Design goals

We aim to design a noise perturbation scheme that achieves the following goals:

- **Utility.** The scheme should not sacrifice the accuracy of the global model. In particular, the FL with the noise perturbation should be able to learn a global model that is as accurate as the regular FL.
- **Dropout-resilience.** The method should handle client dropout due to communication or power failure. When a dropout occurs, the server should still be able to get a reliable aggregation of local model updates from the remaining clients. A limited number of client dropouts should not affect the accuracy of the final FL model.
- **Privacy.** The FL with the scheme should be able to mitigate the inference of private information from the communication of model updates between clients and the server.
- **Efficiency.** The FL with the scheme should not require additional training rounds to achieve a similar accuracy to the regular FL. Additionally, the method should not incur additional computational and communication overhead, since clients are small devices that suffer from limited computational resources and network connectivity in IoT systems.

4. Our approach

4.1. Overview

In light of the drawbacks of DP discussed in Section 2.2, we introduce an adaptive noise scaling method and a direction-based filtering method in the additive noise perturbation scheme. In each iteration, our approach follows the three general steps of FL discussed in Section 2.1. Our approach is similar to FL with the DP scheme in [8,30]. The difference lies in the second step. Instead of sending the original model updates, clients send the noise-perturbed local model updates to the central server, where the noise is generated randomly and locally (see Fig. 2). Our approach is different from the DP scheme in generating random noise. Specifically, a clipping bound is required in DP to limit the influence of a single client. The choice of the clipping bound could have a decisive impact on the utility and privacy of the model. A low clipping bound could destroy the direction of the gradients, weakening its strength in descent of the global model, whereas a high clipping bound might introduce too much noise to the FL system, resulting in an accuracy degradation of the global model. Ideally, the clipping bound should be able to track the change of the norm of the model updates. But practically the behavior of the norms of model updates varies and is hard to predict. A popular method is to use the median of the norms of the unclipped local model updates over the course of training. However, the norm of model updates decreases along the training, whereas the clipping bound may not react as fast as the norm changes. This may introduce excessive noise to the global model, and

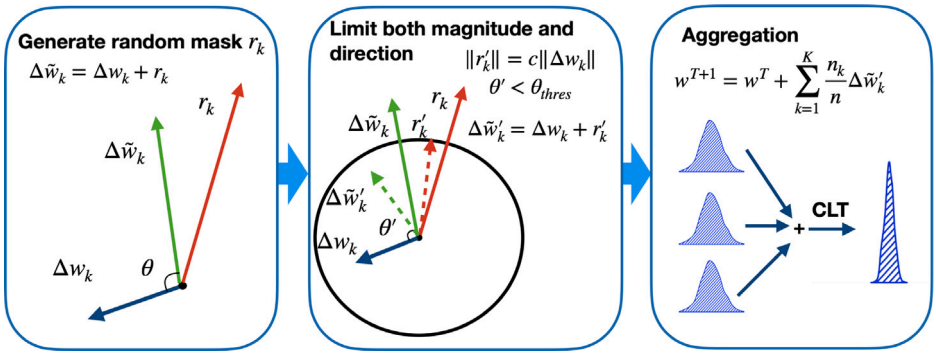


Fig. 2. Geometric illustration of our proposed additive noise perturbation scheme.

this excessive noise could be the cause of the accuracy loss in the global model.

Furthermore, the direction of the gradients plays a significant role in both privacy and model utility aspects and is not considered in the DP scheme. On the one hand, there is plenty of privacy in the direction of the gradients. As indicated in [48], the presence of a given label class can be inferred by analyzing the signs of gradients. Therefore, the noise vector must be well chosen to hide the direction of the gradients. On the other hand, large-scale noise could impair the accuracy or even destroy the convergence of the global model. Two noise vectors with the same magnitude could lead to opposite effects. To be specific, the one in the descent direction could be beneficial to the model convergence, while the other in the ascent direction could destroy the global model convergence.

Our approach is able to achieve a better convergence performance than DP due to the following three features:

- As will be shown in Section 5, setting the magnitude of additive noise to be proportional to the magnitude of local model updates ensures that the additive noise vanishes with local model updates when the FL model convergence occurs, preventing the FL model accuracy degradation.
- The scaling factor c chosen based on the number of participating clients ensures that the FL model enjoys the same convergence performance as a result of the cancelling out presented in the aggregation of noise on the server by the CLT. c can also be chosen to enable the ability to handle dropout clients.
- The proposed direction-based filtering scheme filters out noise vectors in bad directions, accelerating the convergence of the FL model.

Algorithm 1 Our CTL based FL privacy-preserving scheme

Input: K clients with local training datasets $D_k, k \in [K]$; client learning rate η; number of local iterations t; number of aggregations T; angular distance threshold θ_{thres}.
Output: Global model $\tilde{w}^T$.

Initialization global model weight to w^0.
for $T = 0 : T_{max}$ **do**
  The server synchronizes the latest global model to clients, $w_k^{T,0} \leftarrow \tilde{w}^T$.
  for $k = 1 : K$ **do**
    for $\tau = 0 : t-1$ **do**
      The client updates the local weight by $w_k^{T,\tau+1} \leftarrow w_k^{T,\tau} - \eta\nabla F_k(w_k^{T,\tau})$
    end for
    while $\theta < \theta_{thres}$ **do**
      Generate new random noise r_k^T from $\mathcal{N}(0, I)$, and scale them by $\max(1, \frac{c\|\Delta w_k^T\|}{\|r_k^T\|})$, where $\Delta w_k^T = \sum_{\tau=0}^{t-1} \eta\nabla F_k(w_k^{T,\tau})$ and c is a scaler.
      Calculate the angular distance θ from the cosine similarity $\cos(\Delta w_k^{T,\tau}, \Delta w_k^{T,\tau} + r_k^T)$.
    end while
    Add the noise to the local model update, $\Delta\tilde{w}_k^{T,\tau} \leftarrow \Delta w_k^{T,\tau} + r_k^T$.
  end for
  The server aggregates the local model updates from clients, $\Delta\tilde{w}^{T+1} = \sum_{k=1}^{K} \frac{n_k}{n} \Delta\tilde{w}_k^T$, and update the global model $\tilde{w}^{T+1} \leftarrow w^T + \Delta\tilde{w}^{T+1}$.
end for

4.2. Our additive noise scheme

Algorithm 1 details the steps in our proposed noise perturbation scheme, which consists of two key components: the adaptive noise scaling step and the direction-based noise filtering step.

4.2.1. Adaptive noise scaling

We introduce the steps to generate the proposed noise perturbation r_k, and how to determine the value of c in both dropout and non-dropout scenarios. After the client completes the local training, the noise r_k is randomly generated from $\mathcal{N}(0, I)$. Δw_k is denoted as local model updates. Then r_k is scaled by $\frac{c\|\Delta w_k\|}{\|r_k\|}$. The impact of c on model convergence will be theoretically analyzed and numerically evaluated in Sections 5 and 7, respectively.

Determine the value of c in a non-dropout scenario. As indicated in Theorems 2 and 3 (provided later in Section 5), setting the magnitude of additive noise in accordance with the magnitude of local model updates ensures the noise vanishes with the local updates when convergence occurs, avoiding accuracy degradation of the global model. Furthermore, as indicated in Theorem 1 (provided later in Section 5), the standard deviation of the aggregated noise on the server is inversely proportional to the number of participating clients K, indicating that the effect of the scaling factor c will be counteracted by K when aggregated on the server. For convex optimization algorithms (e.g., gradient descent and proximal quasi-Newton), in which the loss function descends in every iteration, the magnitude of additive noise aggregation must not exceed the magnitude of model update aggregation, that is, $\|\sum_{k=1}^{K} r_k\| \leq \|\sum_{k=1}^{K} \Delta w_k\|$. Therefore, in a non-dropout scenario, K is a conservative upper bound for c, i.e., $c \leq K$. For optimization algorithms without monotonic requirement, e.g. SGD, the global model still converges as long as the descent of the global loss function is frequently achieved, indicating that c could be slightly greater than K ($c = K + \epsilon$, where $\epsilon \in \mathbb{N}^+$), which is denoted by $c <_\epsilon K$.

Determine the value of c in a dropout scenario. In a scenario with d client dropouts, the central server is expected to be able to get a reliable aggregation from the remaining $K - d$ clients. FL with our approach can tolerate at most d client dropouts by setting $c <_\epsilon$

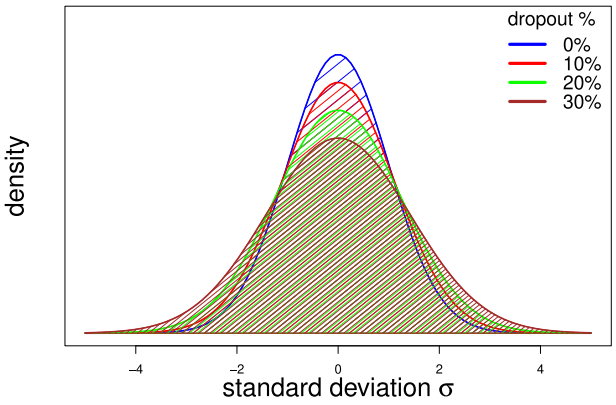


Fig. 3. The distribution of aggregated noise with different dropout probabilities.

$(K - d)$, as previously indicated. Note that c controls the privacy protection strength on clients. Setting a large d results in a reduced c, which also reduces the strength of privacy protection for clients. When there are more than d client dropouts, the distribution of the noise aggregation becomes wider and there will be more noises falling in the tails of the distribution, which could cause the loss function to decrease less frequently (shown in Fig. 3). In particular, when there is an extra dropout of clients, the standard deviation of noise aggregation increases slightly and becomes $\frac{K}{K-1}$ times the standard deviation of noise aggregation of K clients. Therefore, for a sufficiently large K, the impact of a small number of additional client dropouts is limited. There is still a great chance that the server can get a reliable aggregation from the remaining clients.

4.2.2. Direction-based noise filtering

Considering the noise scale alone is insufficient. To limit the negative impact on the accuracy of the FL model, we use cosine similarity to measure the angular distance between the true local model updates and the noise-perturbed local model updates. The client only adds a noise vector whose angular distance is less than the user-defined threshold θ_{thres}. A smaller θ_{thres} leads to a higher chance of global convergence, while a larger θ_{thres} provides better privacy protection.

Note that realistically the dimension of a neural network's parameter vector is usually extremely high. As illustrated in Fig. 4, the angular distance between two arbitrary vectors is Gaussian distributed and becomes more concentrated as the dimension increases. Especially in an extremely high-dimensional space, such as the space of model updates, any two random vectors are orthogonal. Due to this observation, for a fixed θ_{thres}, it could be extremely computationally expensive or even impossible to find a satisfying noise vector in such a high-dimensional space. An intuitive way is to partition the model updates into smaller vectors and apply random noise individually. For convenience, we partition model updates by layers, and noises are generated and added to each layer separately. However, this could raise another problem that setting an absolute value of θ for all layers could be inappropriate. To align θ_{thres} in each layer, we use the three-sigma rule of thumb, setting $\theta_{thres} = \bar{\theta} + \rho\sigma_\theta$, where $\bar{\theta}$ and σ_θ are the mean and standard deviation of θ, respectively, and ρ is the multiple of σ_θ. $\bar{\theta}$ and σ_θ are only related to the dimension of vectors and can be pre-calculated, so this operation does not increase the computational cost. More importantly, this transforms the choice of an absolute value of θ_{thres} into a relative value ρ, in which θ_{thres} is self-adjusted by the dimension of each layer.

The use of a larger c should combine with a small ρ to accelerate FL convergence. However, a smaller ρ increases the similarity between noise-perturbed model updates and original model updates, resulting in less privacy. Also, it could take more time to find a satisfying noise vector for a smaller ρ. Therefore, ρ should be chosen combining privacy requirements according to applications, as well as the choice of c. The numerical results of choosing different settings for ρ will be presented in Section 7.

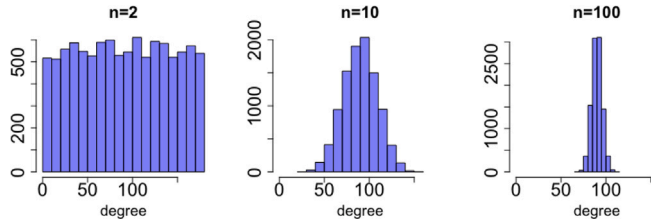


Fig. 4. Histogram of the angular distance (in degree) between two arbitrary vectors in 2, 10 and 100 dimensional spaces, respectively (based on 10,000 samples).

5. Theoretical analysis of our approach

In this section, we study the convergence performance of the proposed perturbation scheme for both convex and non-convex loss functions. The proofs show that FL with our proposed perturbation scheme can achieve the same global model convergence rate and accuracy as that of a regular FL in the convex case, and the same convergence rate as that of a regular FL in the non-convex case.

5.1. Assumptions

Denote the optimal value for $F(\cdot)$ by F^*, and the optimal value for $F_k(\cdot)$ by F_k^*. Define Γ as a measurement of non-i.i.d.-ness across clients: $\Gamma \triangleq \sum_{k=1}^{K} \frac{n_k}{n} F_k^* - F^*$, where $\Gamma \geq 0$ indicates how non-i.i.d. across the client's data. Note that given a large enough number of data samples on clients, we have $\Gamma \to 0$ for i.i.d. data distributions.

Four common assumptions are considered to facilitate the theoretical analyses of our proposed noise perturbation scheme.

Assumption 1. The loss functions $F_k(\cdot)$ for $k \in [K]$ are all L-smooth; that is, $\forall v, w \in \mathbb{R}^d$,

$$F_k(v) - F_k(w) \leq \langle v - w, \nabla F_k(w) \rangle + \frac{L}{2}\|v - w\|^2, \forall k \in [K]. \tag{5}$$

Assumption 2. The loss functions $F_k(\cdot)$ for $k \in [K]$ are all μ-strongly convex; that is, $\forall v, w \in \mathbb{R}^d$,

$$F_k(v) - F_k(w) \geq \langle v - w, \nabla F_k(w) \rangle + \frac{\mu}{2}\|v - w\|^2, \forall k \in [K]. \tag{6}$$

Assumption 3. The expectation of the squared ℓ_2 norm of the stochastic gradients is bounded; that is,

$$\mathbb{E}_\xi \left[\|\nabla F_k(w_k^{T,\tau}, \xi_k^{T,\tau})\|^2 \right] \leq G^2, \ \forall \tau \in [t], |\forall k \in [K]. \tag{7}$$

Assumption 4. For the mini-batch $\xi_k^{T,\tau}$, we have the following.

$$\mathbb{E}_\xi [\nabla F_k(w_k^{T,\tau}, \xi_k^{T,\tau})] = \nabla F_k(w_k^{T,\tau}), \tag{8}$$

where $\mathbb{E}_\xi$ denotes the expectation against the randomness of the stochastic gradient.

5.2. Convergence analysis

We present the following theorems to show the theoretical convergence analyses of FedAvg with our proposed noise perturbation scheme. For simplicity of convergence analysis, we assume that there is no transmission error between the clients and the central server.

For ease of presentation, we denote the noise aggregated on the central server by $R = \sum_{k=1}^{K} \frac{n_k}{n} r_k$, and σ_k denotes the standard deviation of the local additive noise in each element of r_k. We have $\sigma_k \propto c$ and $\sigma_k \propto \Delta w_k$. For simplicity, we also assume that each client has the same amount of data, e.g. $\frac{n_k}{n} \approx \frac{1}{K}$.

Theorem 1. *For a sufficiently large K, each element in R follows the Gaussian distribution $\mathcal{N}(0, \sum_{k=1}^{K} \frac{(\sigma_k)^2}{K^2})$.*

Proof. See Appendix A.

Remark 1. Theorem 1 reveals important properties about the number of participants K and the variance of noise aggregation. (1) The aggregation of additive noise can be characterized by a Gaussian distribution. (2) For sufficiently large K, that is, $K \geq 30$, the contribution of noise on a single client to the variance of aggregation of noise is arbitrarily small. (3) A larger noise scale on the client will result in a greater variance in the aggregation of noise on the server.

The Strongly Convex Case. We analyze the convergence property of our proposed noise perturbation scheme under strong convexity.

Theorem 2. *For a smooth and strongly convex objective function F_k, FedAvg satisfies*

$$\mathbb{E}\left[\ \|\tilde{w}^{T+1} - w^*\|^2\right] \leq A^T \mathbb{E}\left[\|\tilde{w}^0 - w^*\|^2\right] + \sum_{i=0}^{T-1} A^i B \tag{9a}$$

$$A = 2 - \mu\eta t + \mu\eta^2 t \tag{9b}$$

$$B = 2\eta t \Gamma + (1 + 2t)t\eta^2 G^2 (1 + \mu(1 - \eta)) + \frac{t(t+1)(2t+1)\eta^2 G^2}{6} + \frac{9m^2}{K^2} \sum_{k=1}^{K} (\sigma_k^T)^2. \tag{9c}$$

Proof. See Appendix B.

Remark 2. Since $\sigma_k^T \propto c$, we note that B is an increasing function of the noise scale c, while decreasing with the number of participants K. Furthermore, more non-i.i.d. local distributions between clients, resulting in higher Γ and G, will pose a negative impact on the convergence bound.

Remark 3. The FL converges iff $A < 1$, that is, $\eta \in \left[\frac{1-\sqrt{1-\frac{4}{\mu t}}}{2}, \frac{1+\sqrt{1-\frac{4}{\mu t}}}{2}\right]$. Let $\eta = \frac{1}{\sqrt{T}}$ for sufficiently large T and $\eta \in \left[\frac{1-\sqrt{1-\frac{4}{\mu t}}}{2}, \frac{1+\sqrt{1-\frac{4}{\mu t}}}{2}\right]$, the FL with our proposed scheme converges at a rate of $\mathcal{O}(1/T)$, which matches a typical SGD on strongly convex loss functions. In B, the noise-related term $\frac{9m^2}{K^2} \sum_{k=1}^{K} (\sigma_k^T)^2$ decreases as the FL model converges, since $\sigma_k \propto \|\Delta w_k\|$. When convergence occurs, where $\lim_{T\to\infty} \|\Delta w_k^T\| = 0$, we have $\lim_{T\to\infty} \frac{9m^2}{K^2} \sum_{k=1}^{K} (\sigma_k^T)^2 = 0$, which indicates that the proposed scheme converges to the same value as the regular FL scheme under strong convexity.

The Non-convex case. For more general cases, in which the objective function is not necessarily convex, convergence to global optima is not guaranteed, so we will only require convergence to a point of vanishing gradients. We prove the following theorem.

Theorem 3. *For a smooth and non-convex objective function F_k, FedAvg satisfies*

$$\min_{T \in [T_{max}]} \mathbb{E}\|\nabla F(\tilde{w}^t)\|^2 \leq \frac{2(F(w^0) - F(\tilde{w}^*))}{(1 + \eta t - 2\eta)T} + \frac{\eta^3 L^2 t(t+1)(2t+1)G^2}{6(1 + \eta t - 2\eta)} + \frac{m^2 L \sum_{k=1}^{K} (\sigma_k^T)^2}{K^2(1 + \eta t - 2\eta)} + \frac{c^2\eta^2 t^2 G^2}{1 - \eta t - 2\eta}. \tag{10}$$

Proof. See Appendix C.

Remark 4. Let $\eta = \frac{1}{\sqrt{T}}$ for a sufficiently large T, Eq. (10) converges at a rate of $\mathcal{O}(1/\sqrt{T})$, which matches an SGD on non-convex loss functions. The noise-related term, $\frac{m^2 L \sum_{k=1}^{K} (\sigma_k^T)^2}{K^2(1+\eta t-2\eta)}$, decreases as the FL converges due to $\sigma_k \propto \|\Delta w_k\|$. Especially when convergence occurs, where $\lim_{T\to\infty} \|\Delta w_k\| = 0$, we have the noise-related term $\lim_{T\to\infty} \frac{m^2 L \sum_{k=1}^{K} (\sigma_k^T)^2}{K^2(1+\eta t-2\eta)} = 0$. Moreover, since $\sigma_k \propto c$, increasing the number of participating clients K or decreasing c will result in a faster convergence rate.

6. Experiment setup

6.1. Dataset

We evaluate our proposed methods on MNIST, a handwritten digit recognition dataset. The dataset contains 60,000 training data samples and 10,000 testing data samples. Each data sample is a square 28×28 pixel image of a single hand-written digit between 0 and 9.

6.2. Evaluation

We evaluate our proposed scheme from both model utility and privacy protection aspects. And we compare our approach with two baselines: (1) non-private FL, in which clients and servers follow standard FL protocol and do not involve any privacy-preserving mechanisms; (2) FL with local DP, in which clients add DP noise to protect the privacy of their local data. As stated in Section 3.2, our goal is to protect clients' local privacy against an honest-but-curious server and eavesdroppers, thus we only consider adding perturbations on the client's side. We compare our proposed scheme with the (ϵ, δ)-DP proposed in [21], which is widely used as a noise pattern on the client's side [29,30]. Specifically, we use a popular choice of $\sigma = \sqrt{2\log\frac{1.25}{\delta}}/\epsilon$ with a fixed δ of 10^{-5}. The clipping bound is set as the median of the norms of the unclipped local model updates over the course of training.

We evaluate the effectiveness by experimenting with FL with our approach and DP against two state-of-the-art FL privacy inference attacks that we have introduced in Section 2.2: the membership inference attack and the label composition inference attack. The convergence and security performance of our proposed perturbation scheme are evaluated using the following four metrics.

1. **Global model accuracy and convergence rate.** We measure the global model accuracy under different choices of parameters c and ρ as a function of the training epoch, and compare the convergence behavior in both dropout and non-dropout scenarios.
2. **Membership inference attack accuracy and F_1-score.** The attack accuracy is defined as the percentage of data samples that are correctly predicted to be presented in the training dataset. And the F_1 score combines precision and recall into a single value, which is defined as

   $$F_1\text{-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

   A lower accuracy or F_1-score indicates a better protection of privacy.
3. **Accuracy of the label composition inference attack.** The accuracy is measured by the ℓ_2 difference between the true label composition and the inferred label composition. A larger difference indicates better privacy protection.
4. **Signal-to-noise ratio (SNR).** SNR is a popular metric to quantify the relative amount of noise added to the data:

   $$\text{SNR} = \frac{\text{variance of actual data}}{\text{variance of noise}}$$

   A lower SNR indicates that there is a greater amount of noise being introduced into the system, leading to better privacy protection. The recovery of the original data becomes erroneous as the SNR drops below 1 [49]. It is also claimed in [50] that privacy can be achieved without affecting learning performance if a small SNR is consistently achieved.

6.3. FL system settings

We implement FL and privacy inference attacks using the PyTorch framework. We conduct our experiments on Google Colab Pro (CPU: Intel(R) Xeon(R) CPU @ 2.20 GHz; RAM: 13 GB; GPU: Tesla P100-PCIE-16 GB with CUDA 11.2).

The dataset is allocated to 100 clients. The model on each client consists of two convolutional layers and two fully connected layers. In each global training epoch, K clients are randomly selected for the aggregation of the FL model. We use the Dirichlet distribution [51] with the hyper-parameter α to generate different data distributions across clients, in which a smaller α denotes a higher non-i.i.d. level. We set $\alpha = 1$ in the experiments of convergence and membership inference attack, and $\alpha = 0.1, 1, 10, \infty$ in the experiments of label composition inference attack.

For the convergence evaluation, we train the local model with a mini-batch gradient descent with batch size 128, internal epoch $t = 5$, and learning rate $\eta = 0.1$. Ten shadow models and an auxiliary dataset with 3,000 samples are used in the membership inference attack. The training data composition inference attack is launched on local model updates with full-batch gradient descent. To fairly compare our approach with DP, we choose ϵ in DP such that the accuracy is comparable with that of our approach.

7. Experimental results

Our approach achieves the security goals. Recall that we have four security goals (discussed in Section 3.2): utility, resilience to dropout, privacy and efficiency. Our results show that our approach achieves the four goals.

7.1. Utility

The utility of the model is evaluated in the scenario where there is no attack. We fix $\rho = 0$ and choose the scaling factor of our approach to be $c = 1, 3, 5, 10$ and $\epsilon = 15, 20, 30$ in DP and compute the model accuracy as a function of global training epochs. We also include the regular FL to serve as a baseline. As shown in Fig. 5, the trend and final accuracy of our approach are similar to those of the regular FL. For all chosen c, the global model converges to the same accuracy as the regular FL. Such results are in line with Remark 3. Even for large c (e.g., $c = 10$ means that the magnitude of the additive noise is 10 times the magnitude of original model updates), the accuracy curve suffers from slight fluctuations and still achieves the same value as the regular FL does. As the value of c increases, convergence slows slightly due to the increased variance introduced into the global model. This is consistent with our finding in Remark 1. We also plot the global model accuracy w.r.t. ρ, shown in Fig. 6, where the FL model converges to the same value, but faster with a smaller ρ.

Compared with our approach, DP has a different convergence trend, in which convergence is notably slower and it takes more epochs to reach an accuracy comparable to our approach. FL with our approach converges at epoch 5, while DP starts to converge at epoch 10 and the accuracy finally reaches a comparable accuracy at epoch 50 by $\epsilon = 30$.

The final accuracy of the FL model with our approach and DP is presented in Table 1. It is suggested that the training accuracy only drops around 1% as we increase c from 5 to 15 in our approach. It is also indicated that $\epsilon = 30$ is a minimum privacy budget to enable the DP to achieve a similar accuracy to that of $c = 15$ in our approach, as $\epsilon = 20$ reported in the table results in reduced model accuracy. For fairness, we compare under the setting, in which a comparable model accuracy (96%) is achieved by our approach ($c = 15$) and DP ($\epsilon = 30$).

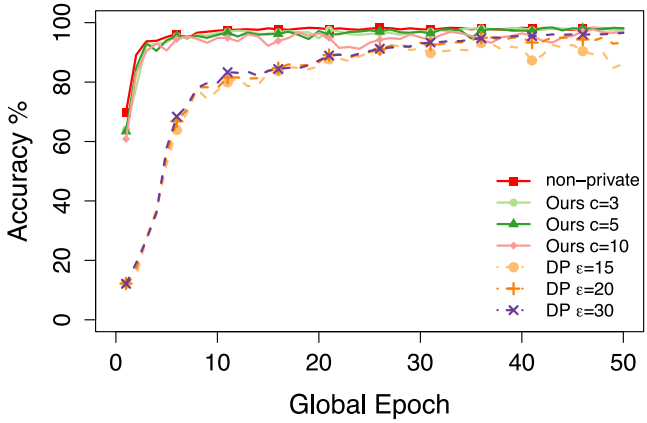


Fig. 5. Comparison of the model accuracy among the non-private FL, FL with our perturbation scheme, and FL with DP.

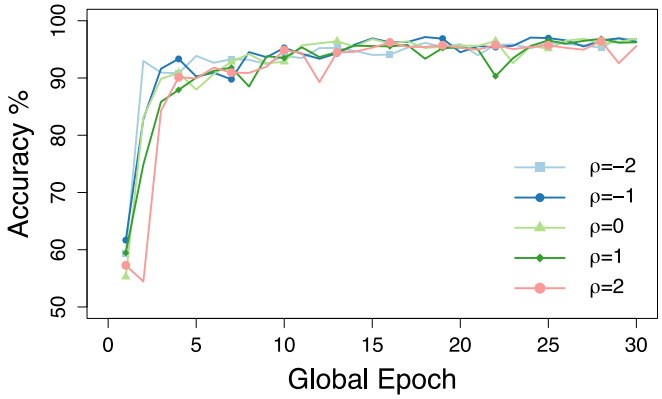


Fig. 6. The accuracy of the FL model with different ρ.

Table 1
Accuracy of the FL model with our approach, FL with DP, and regular FL.

	Regular	DP			Our approach		
		$\epsilon = 15$	$\epsilon = 20$	$\epsilon = 30$	$c = 5$	$c = 10$	$c = 15$
Accuracy (%)	98.26	86.15	93.28	96.56	98.08	97.1	96.92

7.2. Dropout-resilience

In Section 5, we have shown that our approach can handle up to d client dropouts by setting $c <_\epsilon (K - d)$. Therefore, in this scenario, the convergence performance is similar to that of the non-dropout scenario where we have $c <_\epsilon K$. We also investigate the convergence performance when there are additional client dropouts. In particular, each client has a dropout probability from 0% (non-dropout) to 40%. And we set $c = 15$ in our approach and $\epsilon = 30$ in DP. When dropout occurs, the server will experience an increased variance of the aggregated noise, which might impair the global model's convergence and accuracy. As shown in Fig. 7, as the dropout probability increases from 10% to 40%, the global model convergence rate and the accuracy of our approach remain similar to that of the non-dropout case. Our theoretical findings in Remark 2 are consistent with these experimental results. Reducing a limited number of participating clients does not affect the global model accuracy, but only results in a slightly slower convergence. As for DP, both the global model convergence rate and the accuracy are severely impacted. Therefore, our approach handles up to d client dropouts by setting $c <_\epsilon (K - d)$, and the convergence performance of the global model is stable even with additional client dropouts.

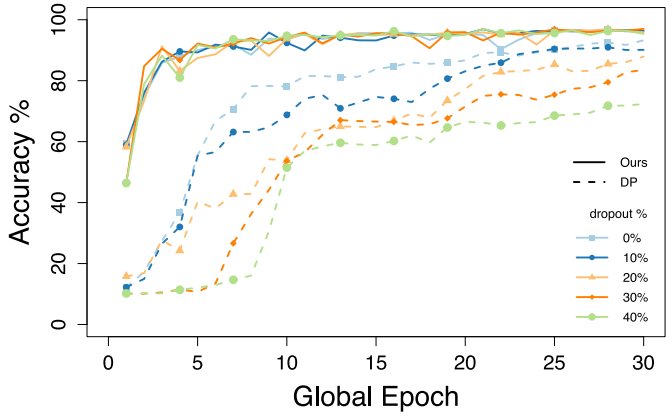


Fig. 7. The FL model accuracy w.r.t. dropout probability.

Table 2
Membership inference attack accuracy and F_1 score for regular FL, FL with our approach, and FL with DP with different ϵ and c.

	Regular	DP		Our approach		
		$\epsilon = 20$	$\epsilon = 30$	$c = 5$	$c = 10$	$c = 15$
Attack accuracy (%)	87.36	79.9	80.12	80.48	80.03	79.82
Attack F_1 score	0.87	0.52	0.57	0.61	0.52	0.53

Table 3
Membership inference accuracy and F_1-score of our approach with $c = 20$ and different ρ.

	Our approach				
	$\rho = 2$	$\rho = 1$	$\rho = 0$	$\rho = -1$	$\rho = -2$
Main accuracy (%)	94.76	95.26	96.92	97.56	98.10
Attack accuracy (%)	79.12	79.75	79.82	78.84	79.72
Attack F_1-score	0.49	0.50	0.53	0.52	0.53

7.3. Privacy

7.3.1. Defending against membership inference attack

We continue to use the setting of $c = 10, 15$ in our approach and $\epsilon = 20, 30$ in DP. Fig. 8 shows the per-class attack accuracy and F_1-score of the membership inference attack against FL with our approach, FL with DP, and regular FL. As expected, regular FL leaks a considerable amount of information about the training dataset, resulting in an attack success rate as high as 87% on average. Both DP and our approach can reduce attack accuracy and the F_1 score against the membership inference attack. There are no significant differences in attack accuracy. Regarding the F_1-score, authors in [33] set the baseline F_1-score to 0.67 (dotted line in Fig. 8(b)), since there are equal numbers of members and nonmembers in the attack test dataset. The F_1-scores of all private models are below the baseline. The F_1-score for DP with $\epsilon = 30$ presents a higher pattern, whereas there is no significant difference among the rest of privacy-preserving FL models.

Furthermore, Table 2 indicates that our approach with $c = 10, 15$ is as effective as DP with $\epsilon = 20$. Referring back to Table 1, we see that the accuracy of FL with $\epsilon = 20$ in DP is 3% less than FL with $c = 10, 15$ in our approach. Therefore, given the same strength to defend against the membership inference attack, FL with our approach achieves a higher global model accuracy.

Furthermore, Table 3 provides the global model accuracy, attack accuracy, and F_1 score for a fixed $c = 15$ and different value of ρ. It is suggested that increasing ρ results in slightly decreased accuracy of the FL model, but greater privacy protection in terms of the F_1 score.

Table 4
Time complexity of different ρ values in terms of multiples of that of DP.

Value of ρ	3	2	1	0	−1	−2	−3
Multiples of DP cost (m_ρ)	1.0	1.0	1.1	2.0	6.3	43.9	333.3

7.3.2. Defending against label composition inference attack

To compare privacy protection in different local label composition scenarios, we consider four local distribution settings, including an i.i.d. ($\alpha = \infty$) and three non-i.i.d. local distribution settings ($\alpha = 10, 1, 0.1$). Fig. 9 visualizes the label composition with different α.

The results of the label composition inference attack are presented in Fig. 10, which shows a box plot of the ℓ_2 distance between the original label composition and the inferred label composition of FL with our approach and DP. Our approach is more effective in defending the distribution inference attack compared with DP as the local distribution becomes more i.i.d ($\alpha = \infty, 10$), whereas our approach and DP achieve comparable protection as local distributions become more dissimilar ($\alpha = 1, 0.1$).

7.3.3. Signal-to-noise ratio (SNR)

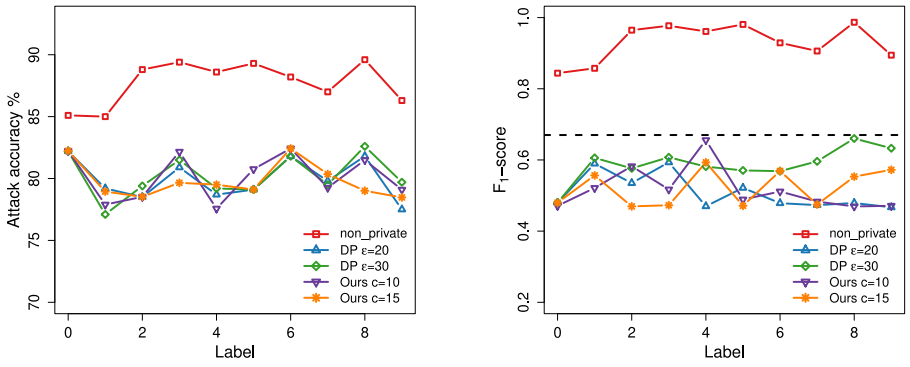
Finally, we present the SNR of FL with our approach and DP as a function of the training epochs in Fig. 11. Similarly as in previous experiments, the ϵ for DP and the c in our approach are chosen such that a similar global model accuracy is achieved. The results show that the SNR of DP is high at the beginning of the training and decreases as the convergence occurs, while our approach achieves a consistently low SNR. Referring to [50], such a consistently low SNR also explains our results in Section 7.1 that our approach has a minor impact on the global model's convergence and accuracy.

Furthermore, the results in [49] showed that the original data could be more difficult to recover from a lower SNR. As shown in Fig. 11, FL with DP has a higher chance of recovery in the early training stage, due to their higher SNR values.

7.4. Efficiency

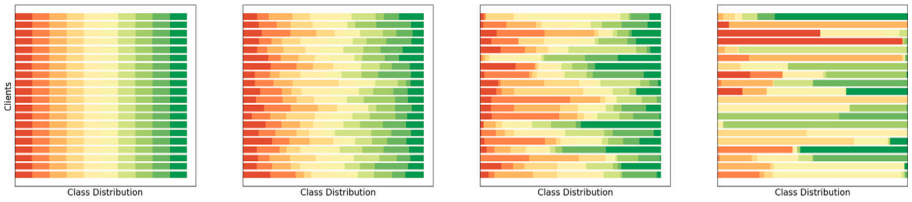
We analyze the efficiency based on both communication and computational overhead. FL with our approach converges as fast as regular FL and much faster than FL with DP. Especially, for the MNIST task, both the FL with our approach and the regular FL converge at epoch 5, but FL with DP requires extra epochs to reach a similar global model accuracy, indicating that extra communication is needed for DP. Therefore, the communication overhead of our proposed scheme is similar to that of the regular FL and much lower than that of the DP.

Furthermore, compared with the regular FL, the only additional computational cost of our approach lies in random noise generation, specifically direction-based filtering. Table 4 shows the time complexity analysis of our proposed noise perturbation scheme w.r.t. ρ in terms of the multiples (m_ρ) of DP. In general, the time complexity of our approach is inversely related to ρ. In DP, generating a noise vector for a vector of model updates with n parameters costs $\mathcal{O}(n)$. Therefore, the time complexity of our approach is $m_\rho \times \mathcal{O}(n)$. Since m_ρ is much less than n in practice, the time complexity of our proposed method is still $\mathcal{O}(n)$. The real time spent on generating the noise vector for one client's local updates w.r.t. ρ is presented in Table 5. The time cost shows an increasing pattern with a decreasing value of ρ. Even for a small ρ (e.g., $\rho = -3$), the time spent generating the noise vector is 0.52 s, which is minor compared with the local training time, which is 3 seconds in our experiments. Thus, we claim that our approach does not introduce additional communication and computational overhead.



(a) Attack accuracy. (b) Attack F_1-score.

Fig. 8. Per-class accuracy and F_1 score of the membership inference attack against FL with DP, FL with our perturbation scheme, and non-private FL.



(a) $\alpha = \infty$ (b) $\alpha = 10$ (c) $\alpha = 1$ (d) $\alpha = 0.1$

Fig. 9. Local label composition w.r.t. α.

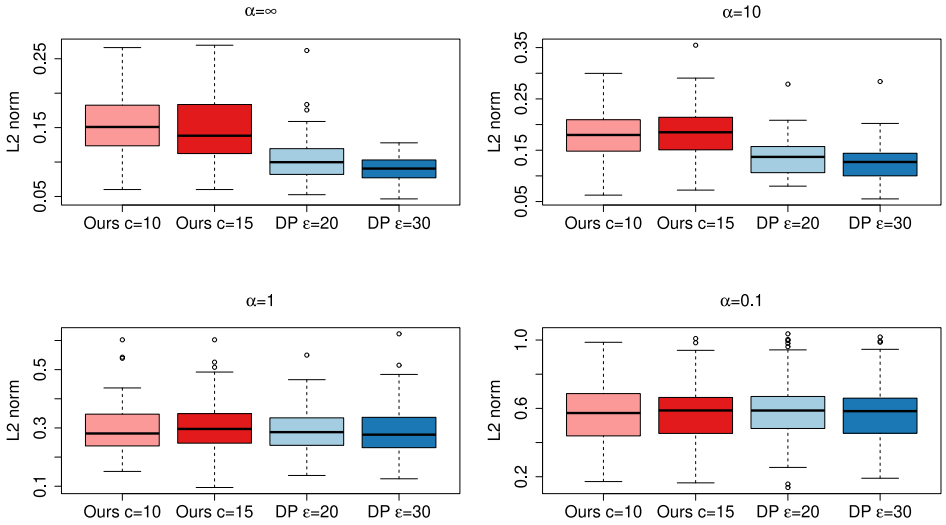


Fig. 10. Box plot of the ℓ_2 distance between the original label composition and the inferred label composition by our approach and DP.

Table 5
Real time spent on noise vector generation w.r.t ρ.

	DP	Value of ρ						
		3	2	1	0	−1	−2	−3
Time (s)	0.015	0.018	0.018	0.019	0.021	0.034	0.143	0.520

7.5. Generalization to more complex datasets

To explore whether the above findings still hold for more complex datasets and neural network architectures, we conduct several experiments using ResNet 18 [52] on the CIFAR-10 [53] datasets. CIFAR-10 consists of 60,000 32 × 32 color images containing one of ten object classes, with 6000 images per class. ResNet 18 is a convolutional neural network that is 18 layers deep and contains around 11 million parameters.

The data are distributed to 50 clients with a non-i.i.d. parameter $\alpha = 10$ and 10 clients are selected in each training round. We use SGD with a learning rate of 0.1 and an epoch of 200. We compare FL with the proposed method ($c = 10$) with non-private FL and FL with DP ($\epsilon = 100$ and $\delta = 10^{-5}$). We report the training accuracy, the attack accuracy and F_1-score of the membership inference attack, and the accuracy of the label composition inference attack. These experiments are representative in verifying the impact of our proposed method on FL convergence and accuracy, and the privacy protection against state-of-the-art privacy inference attacks.

The accuracy of the FL model is presented in Fig. 12. The FL with the proposed method converges slightly slower than the non-private FL,

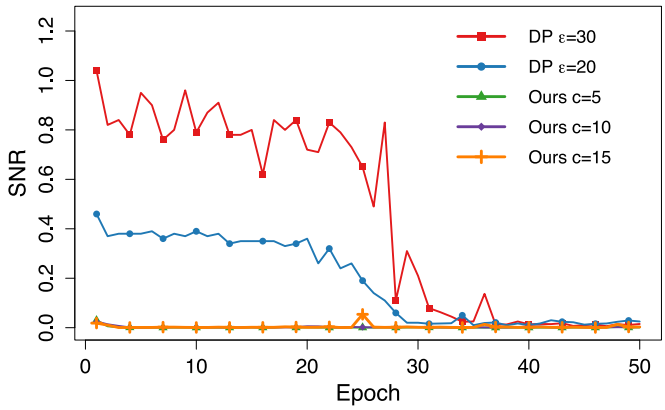


Fig. 11. The SNR of our approach and the DP during the training course.

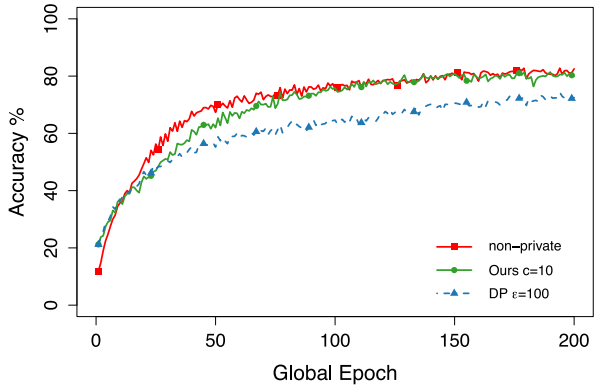


Fig. 12. Global model accuracy of non-private FL, FL with DP, FL with our method on CIFAR-10, respectively.

Table 6
FL model accuracy, overall attack accuracy and F_1-score, and mean ℓ_2 distance on CIFAR-10.

	Non-private	Ours $c = 10$	DP $\epsilon = 100$
Model accuracy (%)	82.54	81.16	74.30
Attack accuracy (%)	76.77	63.06	62.36
Attack F_1-score	0.871	0.527	0.526
Attack ℓ_2 distance	0.023	0.101	0.099

but still converges to a similar accuracy of 80% around epoch 100. The slower convergence rate is due to higher non-convexity in the ResNet 18 model, which is consistent with the convergence analysis of the non-convex case (Remark 3). For FL with DP, even for a large ϵ of 100, the FL model still suffers from accuracy loss and can only reach an accuracy of 74%.

We continue to evaluate the effectiveness of privacy protection on CIFAR-10. Table 6 summarizes the FL model accuracy and overall attack accuracy and F_1-score against the membership inference attack, as well as the ℓ_2 distance against the label composition inference attack. More specifically, Fig. 13 provides the per-class attack accuracy and F_1-score. Fig. 14 presents the results for the label composition inference attack, which shows a box plot of ℓ_2 distance of the true label composition and the inferred ones. Compared to non-private FL, both DP and our method can significantly lower the strength of two attacks, since the accuracy of the attack, F_1 score, and the ℓ_2 distance are reduced by 17% percent, 0.3 and 0.078, respectively. There is no significant difference between our method and DP in both per-class attack accuracy and attack F_1-score, as well as the attack ℓ_2 norm. However, the gain in privacy protection by DP comes at the cost of 8% model accuracy loss, while our method enjoys a lossless accuracy.

8. Conclusion and future work

In this paper, we have proposed a novel adaptive perturbation-based scheme that protects local privacy in FL but without sacrificing the accuracy of the global model. The key difference between our approach and differential privacy is that we considered both magnitude and direction when generating random noise. In particular, we introduced adaptive noise scaling and direction-based filtering methods to reduce the negative impact of noise on the global model. We have provided theoretical convergence analyses of our proposed scheme with both non-convex and convex FL loss functions. Numerical experiments on the MNIST and CIFAR-10 datasets have shown that our approach can achieve a convergence performance comparable to that of the regular FL. And our proposed noise perturbation scheme can achieve comparable, or in many cases, stronger privacy protection than DP in defending against state-of-the-art membership inference attack and label composition inference attack.

Although FL combined with privacy-preserving methods has made great progress in protecting data privacy, there is still a gap between FL techniques and real IoT applications, where the key challenges in IoT systems come from computational and power constraints. Due to the heterogeneity of IoT devices, privacy budgets can differ between devices or even between data samples on a single device. Future research should focus on reducing computational and communication overhead, preserving model accuracy, and enabling the ability to handle mixed privacy constraints. In particular, two different branches deserve further investigation. First, a noise tolerance bound could be derived in the scenario where each client has their own privacy budget using the generalized central limit theorem. Second, the privacy budget could be taken into account when determining the optimal aggregation interval and the number of participants to trade off training time and communication overhead.

CRediT authorship contribution statement

Tian Liu: conceptualization, Methodology, Theoretical proof, Experiments and results interpretation, Writing – original draft. **Xueyang Hu:** Proof and results interpretation. **Hairuo Xu:** Methodology, Writing – reviewing and editing. **Tao Shu:** Supervision. **Diep N. Nguyen:** Writing – reviewing and editing.

Declaration of competing interest

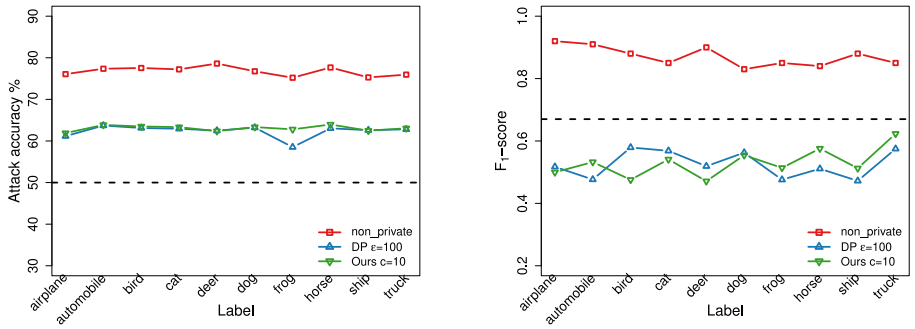
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported in part by the United States National Science Foundation (NSF) under grants CNS-2006998 and CNS-1837034. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF.



(a) Attack accuracy. (b) Attack F_1-score.

Fig. 13. Per-class attack accuracy and F_1-score of the membership inference attack against FL with DP, FL with our approach, and regular FL on CIFAR-10. The dotted lines are baselines, where there is no privacy-preserving mechanism.

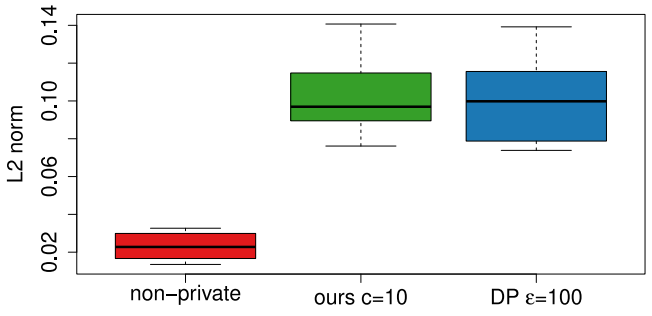


Fig. 14. Box plot of ℓ_2 distance between the true and inferred label composition on CIFAR-10 with non-i.i.d. $\alpha = 10$.

Appendix A. Proof of Theorem 1

Proof. Recall that in the process of generating random noise, r_k is first randomly chosen from $\mathcal{N}(0, I)$ and then scaled by $\frac{c\|\Delta w_k\|}{\|r_k\|}$. Therefore, the i -th element r_{ki} follows a Gaussian distribution $\mathcal{N}(0, \sigma_k^2)$ with $\sigma_k = \frac{c\|\Delta w_k\|}{\|r_k\|}$. For the sequence $\{r_{ki}\}$ for $k \in [K]$, if the Lindeberg's condition holds, then $\frac{1}{K}\sum_{k=1}^{K} r_{ki} \to \mathcal{N}(0, \frac{1}{K^2}\sum_{k=1}^{K}\sigma_k^2)$. Thus, we must verify that for any $\epsilon > 0$,

$$\lim_{K\to+\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}[r_{ki}^2 \cdot \mathbf{1}\{|r_{ki}|^2 \geq \epsilon\sqrt{K}\}] = 0, \tag{A.1}$$

where $\mathbf{1}$ is the indicator function. Note that r_{ki} can be represented by $\sigma_k \cdot x$, where x denotes a standard Gaussian random variable. Then we have the following.

$$\mathbb{E}\big[r_{ki}^2 \cdot \mathbf{1}\{|r_{ki}|^2 \geq \sum_{k=1}^{K}\epsilon\sqrt{K}\}\big] \leq \sigma_k^2\mathbb{E}\big[x^2 \cdot \mathbf{1}\{|x|^2 \geq \sum_{k=1}^{K}\epsilon\sqrt{K}\}\big]. \tag{A.2}$$

And Eq. (A.2) goes to 0 when K is sufficiently large. □

Appendix B. Proof of Theorem 2

This proof is deeply inspired by the proof developed in [54], and we roughly follow the same proof procedure.

Proof. The noise-perturbed global model parameter is updated as

$$\tilde{w}^{T+1} = \tilde{w}^T - \sum_{k=1}^{K}\frac{n_k}{n}\Delta\tilde{w}_k^T + R^T. \tag{B.1}$$

Assuming that w^* is the optimal parameter, we have the following.

$$\begin{aligned}
&\mathbb{E}\big[\ \|\tilde{w}^{T+1} - w^*\|^2\big] \\
&=\mathbb{E}\big[\|\tilde{w}^T - w^*\|^2\big] \underbrace{-2\mathbb{E}\big[\langle\sum_{k=1}^{K}\frac{n_k}{n}\Delta\tilde{w}_k^T, R^T\rangle\big]}_{B_1} + \underbrace{\mathbb{E}\big[\|\sum_{k=1}^{K}\frac{n_k}{n}\Delta\tilde{w}_k^T\|^2\big]}_{B_2} + \underbrace{\mathbb{E}\big[\|R^T\|^2\big]}_{B_3} \\
&+ \underbrace{2\mathbb{E}\big[\langle\tilde{w}^T - w^*, R^T\rangle\big]}_{B_4} \underbrace{-2\mathbb{E}\big[\langle\tilde{w}^T - w^*, \sum_{k=1}^{K}\frac{n_k}{n}\Delta\tilde{w}_k^T\rangle\big]}_{B_5}
\end{aligned} \tag{B.2}$$

Next, we bound the terms on the RHS of (B.2). By Young's inequality, we have $B_1 \leq B_2 + B_3$. By the Cauchy–Schwarz inequality, we have

$$B_2 = \mathbb{E}\big[\big\|\sum_{k=1}^{K}\frac{n_k}{n}\Delta\tilde{w}_k^T\big\|^2\big] \leq \sum_{k=1}^{K}\frac{n_k}{n}\mathbb{E}\big[\big\|\Delta\tilde{w}_k^T\big\|^2\big] \tag{B.3}$$

$$= \eta^2\sum_{k=1}^{K}\frac{n_k}{n}\mathbb{E}\big[\big\|\sum_{\tau=0}^{t-1}\nabla F_k(w_k^{T,\tau}, \xi_k^{T,\tau})\big\|^2\big] \tag{B.4}$$

$$\leq \eta^2 t\sum_{\tau=0}^{t-1}\sum_{k=1}^{K}\frac{n_k}{n}\mathbb{E}\big[\big\|\nabla F_k(w_k^{T,\tau}, \xi_k^{T,\tau})\big\|^2\big] \leq \eta^2 t^2 G^2, \tag{B.5}$$

$$B_3 = \mathbb{E}\big[\|R^T\|^2\big] \leq \frac{9m^2}{K^2}\sum_{k=1}^{K}(\sigma_k^T)^2, \tag{B.6}$$

where m is the dimension of the model parameter and the inequality holds by Theorem 1 for a sufficiently large K. Again, by the Cauchy–Schwarz inequality, we have

$$B_4 = 2\mathbb{E}\big[\langle\tilde{w}^T - w^*, R^T\rangle\big] \leq \mathbb{E}\big[\|\tilde{w}^T - w^*\|^2\big] + B_3. \tag{B.7}$$

$$\begin{aligned}
B_5 =& 2\mathbb{E}\big[\langle w^* - \tilde{w}^T, \sum_{k=1}^{K}\frac{n_k}{n}\Delta\tilde{w}_k^T\rangle\big] \\
\leq& 2\eta\sum_{k=1}^{K}\sum_{\tau=0}^{t-1}\frac{n_k}{n}\mathbb{E}\big[\langle w^* - \tilde{w}^T, \nabla F_k(\tilde{w}_k^{T,\tau}, \xi_k^{T,\tau})\rangle\big] \\
\leq& \underbrace{2\eta\sum_{k=1}^{K}\sum_{\tau=0}^{t-1}\frac{n_k}{n}\mathbb{E}\big[\langle\tilde{w}_k^{T,\tau} - \tilde{w}^T, \nabla F_k(\tilde{w}_k^{T,\tau}, \xi_k^{T,\tau})\rangle\big]}_{C_1} \\
&+ \underbrace{2\eta\sum_{k=1}^{K}\sum_{\tau=0}^{t-1}\frac{n_k}{n}\mathbb{E}\big[\langle w^* - \tilde{w}_k^{T,\tau}, \nabla F_k(\tilde{w}_k^{T,\tau}, \xi_k^{T,\tau})\rangle\big]}_{C_2}
\end{aligned} \tag{B.8}$$

$$\begin{aligned}
C_1 =& \sum_{k=1}^{K}\sum_{\tau=0}^{t-1}\frac{n_k}{n}\mathbb{E}\big[\|\tilde{w}_k^{T,\tau} - \tilde{w}^T\|^2\big] + \eta^2\sum_{k=1}^{K}\sum_{\tau=0}^{t-1}\frac{n_k}{n}\mathbb{E}\big[\|\nabla F_k(\tilde{w}_k^{T,\tau}, \xi_k^{T,\tau})\|^2\big] \\
\leq& \sum_{k=1}^{K}\sum_{\tau=0}^{t-1}\frac{n_k}{n}\mathbb{E}\Big[\Big\|\eta\sum_{i=0}^{\tau}\nabla F_k(\tilde{w}_k^{T,i}, \xi_k^{T,i})\Big\|^2\Big] + \eta^2 tG^2
\end{aligned}$$

$$\leq \frac{t(t+1)(2t+1)\eta^2 G^2}{6} + \eta^2 t G^2, \tag{B.9}$$

$$C_2 \overset{(e)}{\leq} 2\eta \sum_{k=1}^{K}\sum_{\tau=0}^{t-1} \frac{n_k}{n} \mathbb{E}\big[\langle w^* - \tilde{w}_k^{T,\tau}, \nabla F_k(\tilde{w}_k^{T,\tau})\rangle\big] \tag{B.10}$$

$$\begin{aligned}
&\overset{(f)}{\leq} 2\eta \sum_{k=1}^{K}\sum_{\tau=0}^{t-1} \frac{n_k}{n} \mathbb{E}\big[F_k(w^*) - F_k(\tilde{w}_k^{T,\tau}) - \frac{\mu}{2}\|\tilde{w}_k^{T,\tau} - w^*\|^2\big] \\
&\leq 2\eta \sum_{k=1}^{K}\sum_{\tau=0}^{t-1} \frac{n_k}{n} \mathbb{E}\big[F_k(w^*) - F_k^* + F_k^* - F_k(\tilde{w}_k^{T,\tau}) - \frac{\mu}{2}\|\tilde{w}_k^{T,\tau} - w^*\|^2\big] \\
&= 2\eta t\Gamma + 2\eta \sum_{k=1}^{K}\sum_{\tau=0}^{t-1} \frac{n_k}{n}(F_k^* - F_k(\tilde{w}_k^{T,\tau})) - \mu\eta \sum_{k=1}^{K}\sum_{\tau=0}^{t-1} \frac{n_k}{n} \mathbb{E}\underbrace{\big[\|\tilde{w}_k^{T,\tau} - w^*\|^2\big]}_{C_3},
\end{aligned} \tag{B.11}$$

where (e) and (f) are due to Assumptions 2 and 4, respectively.

$$\begin{aligned}
C_3 &= \|\tilde{w}_k^{T,\tau} - \tilde{w}^T\|^2 + \|\tilde{w}^T - w^*\|^2 + 2\langle \tilde{w}_k^{T,\tau} - \tilde{w}^T, \tilde{w}^T - w^*\rangle \\
&\leq \|\tilde{w}_k^{T,\tau} - \tilde{w}^T\|^2 + \|\tilde{w}^T - w^*\|^2 - \frac{1}{\eta}\|\tilde{w}_k^{T,\tau} - \tilde{w}^T\|^2 - \eta\|\tilde{w}^T - w^*\|^2 \\
&= (1-\eta)\|\tilde{w}_k^T - w^*\|^2 - (\frac{1}{\eta} - 1)\|\tilde{w}_k^{T,\tau} - \tilde{w}^T\|^2,
\end{aligned} \tag{B.12}$$

Substituting C_3 into C_2, we have

$$\begin{aligned}
C_2 =& 2\eta t\Gamma + 2\eta \sum_{k=1}^{K}\sum_{\tau=0}^{t-1} \frac{n_k}{n}(F_k^* - F_k(\tilde{w}_k^{T,\tau})) - \mu\eta t(1-\eta)\mathbb{E}\big[\|\tilde{w}^T - w^*\|^2\big] \\
&+ \mu(1-\eta)\eta^2 G^2 \frac{t(t+1)(2t+1)}{6}.
\end{aligned} \tag{B.13}$$

Substituting C_1 and C_2 into B_5, we have

$$\begin{aligned}
B_5 \leq& -\mu\eta t(1-\eta)\mathbb{E}\big[\|\tilde{w}^T - w^*\|_2^2\big] + (1+\mu(1-\eta))\frac{t(t+1)(2t+1)\eta^2 G^2}{6} \\
&+ \eta^2 t G^2 + 2\eta t\Gamma \\
&+ 2\eta \sum_{k=1}^{K}\sum_{\tau=0}^{t-1} \frac{n_k}{n}(F_k^* - F_k(\tilde{w}_k^{T,\tau})).
\end{aligned} \tag{B.14}$$

Substituting $B_1 - B_5$ into Eq. (B.2), we have

$$\begin{aligned}
\mathbb{E}\big[\|\tilde{w}^{T+1} - w^*\|^2\big] \overset{(g)}{\leq}& (2 - \mu\eta t(1-\eta))\mathbb{E}\big[\|\tilde{w}^T - w^*\|_2^2\big] \\
&+ 2\eta t\Gamma + (1+2t)t\eta^2 G^2 \\
&+ (1+\mu(1-\eta))\frac{t(t+1)(2t+1)\eta^2 G^2}{6} \\
&+ \frac{9m^2}{K^2}\sum_{k=1}^{K}(\sigma_k^T)^2,
\end{aligned} \tag{B.15}$$

where (g) follows from $F_k^* - F_k(\tilde{w}_k^{T,\tau}) \leq 0$. Rearranging Eq. (B.15) and summing from 0 to T, we have proved Theorem 2. □

Appendix C. Proof of Theorem 3

Proof. We denote the global model parameter at aggregation T by $\tilde{w}^{T+1} = \tilde{w}^T - \Delta w^T + R^T$, where $\Delta w^T = \eta \sum_{k=1}^{K}\sum_{\tau=0}^{t-1} \frac{n_k}{n} \nabla F_k(\tilde{w}_k^{T,\tau}, \xi_k^{T,\tau})$. Due to the smoothness of Assumption 1 and taking the expectation of $F_k(\tilde{w}^{T+1})$ over randomness at the Tth aggregation, we have

$$\mathbb{E}[F(\tilde{w}^{T+1})] \leq F(\tilde{w}^T) + \langle \nabla F(\tilde{w}^T), \mathbb{E}[R^T - \Delta w^T]\rangle + \frac{L}{2}\mathbb{E}[\|R^T - \Delta w^T\|^2] \tag{C.1}$$

$$\begin{aligned}
&\leq F(\tilde{w}^T) + \langle \nabla F(\tilde{w}^T), \mathbb{E}[R^T - \Delta w^T + \eta\nabla F(\tilde{w}^T) - \eta\nabla F(\tilde{w}^T)]\rangle \\
&\quad + \frac{L}{2}\mathbb{E}[\|R^T - \Delta w^T\|^2]
\end{aligned} \tag{C.2}$$

$$\begin{aligned}
&\leq F_k(\tilde{w}^T) + \underbrace{\langle \nabla F(\tilde{w}^T), \mathbb{E}[\eta\nabla F(\tilde{w}^T) - \Delta w^T]\rangle}_{A_1} \\
&\quad + \underbrace{\frac{L}{2}\mathbb{E}[\|R^T - \Delta w^T\|^2]}_{A_2} \\
&\quad + \underbrace{\frac{1}{2}\mathbb{E}\|R^T\|^2}_{A_3} + \frac{1}{2}\|\nabla F(\tilde{w}^T)\|^2 - \eta\|\nabla F(\tilde{w}^T)\|^2.
\end{aligned} \tag{C.3}$$

$$A_1 = \langle \nabla F(\tilde{w}^T), \mathbb{E}\big[\eta\nabla F(\tilde{w}^T) - \Delta w^T\big]\rangle \tag{C.4}$$

$$= \langle \sqrt{\eta t}\nabla F(\tilde{w}^T), \frac{\sqrt{\eta}}{\sqrt{t}}\mathbb{E}[\sum_{k=1}^{K}\sum_{\tau=0}^{t-1}\frac{n_k}{n}(\nabla F_k(\tilde{w}^T) - \nabla F_k(\tilde{w}_k^{T,\tau}, \xi_k^{T,\tau}))]\rangle \tag{C.5}$$

$$\overset{(b)}{\leq} \frac{\eta t}{2}\|\nabla F(\tilde{w}^T)\|^2 + \frac{\eta}{2t}\mathbb{E}\big[\|\sum_{k=1}^{K}\sum_{\tau=0}^{t-1}\frac{n_k}{n}(\nabla F_k(\tilde{w}^T) - \nabla F_k(\tilde{w}_k^{T,\tau}, \xi_k^{T,\tau}))\|\big]^2 \tag{C.6}$$

$$\overset{(c)}{\leq} \frac{\eta t}{2}\|\nabla F(\tilde{w}^T)\|^2 + \frac{\eta L^2}{2}\sum_{k=1}^{K}\sum_{\tau=1}^{t-1}\frac{n_k}{n}\mathbb{E}\big[\|\tilde{w}_k^{T,\tau} - \tilde{w}^T\|^2\big]$$

$$\leq \frac{\eta t}{2}\|\nabla F(\tilde{w}^T)\|^2 + \frac{\eta^3 L^2}{2}\sum_{k=1}^{K}\sum_{\tau=0}^{t-1}\frac{n_k}{n}\mathbb{E}\|\sum_{i=0}^{\tau}\nabla F_k(\tilde{w}_k^{T,i}, \xi_k^{T,i})\|^2 \tag{C.7}$$

$$\overset{(d)}{\leq} \frac{\eta t}{2}\|\nabla F(\tilde{w}^T)\|^2 + \frac{\eta^3 L^2 t(t+1)(2t+1)}{12}G^2, \tag{C.8}$$

where (b) follows from the Young inequality, and (c) is due to Assumption 1 and $\mathbb{E}\|\sum_{i=1}^{n} x_i\|^2 \leq n\sum_{i=1}^{n}\mathbb{E}\|x_i\|^2$, and (d) is due to Assumption 3.

Based on the relationship of the noise and the gradient and following the Efron-Stein inequality, we have

$$A_2 = \frac{L}{2}\mathbb{E}[\|R^T - \Delta w^T\|^2] \leq \frac{m^2 L}{2K^2}\sum_{k=1}^{K}(\sigma_k^T)^2, \tag{C.9}$$

where m is the dimension of r_k.

$$A_3 = \frac{1}{2}\mathbb{E}\big[\|R^T\|^2\big] \leq \frac{1}{2}c^2\eta^2\mathbb{E}\big[\|\sum_{k=1}^{K}\sum_{\tau=0}^{t-1}\frac{n_k}{n}\nabla F_k(\tilde{w}_k^{T,\tau}, \xi_k^{T,\tau})\|\big] \tag{C.10}$$

$$\leq \frac{1}{2}c^2\eta^2\sum_{k=1}^{K}\frac{n_k}{n}\mathbb{E}\big[\|\sum_{\tau=0}^{t-1}F_k(\tilde{w}_k^{T,\tau}, \xi_k^{T,\tau})\|\big] \tag{C.11}$$

$$\leq \frac{1}{2}c^2\eta^2 t\sum_{k=1}^{K}\frac{n_k}{n}\sum_{\tau=0}^{t-1}\mathbb{E}\big[\|F_k(\tilde{w}_k^{T,\tau}, \xi_k^{T,\tau})\|\big] \tag{C.12}$$

$$\leq \frac{1}{2}c^2\eta^2 t^2 G^2 \tag{C.13}$$

Substituting A_1, A_2, and A_3 into Eq. (C.8), we have

$$\begin{aligned}
\mathbb{E}[F(\tilde{w}^{T+1})] \leq& F(\tilde{w}^T) + (\frac{1+\eta t - 2\eta}{2})\|\nabla F(\tilde{w}^T)\|^2 + \frac{\eta^3 L^2 t(t+1)(2t+1)}{12}G^2 \\
&+ \frac{m^2 L}{2K^2}\sum_{k=1}^{K}(\sigma_k^T)^2 + \frac{1}{2}c^2\eta^2 t^2 G^2.
\end{aligned} \tag{C.14}$$

Rearranging Eq. (C.14) and summing from $0 - T$, we have

$$\begin{aligned}
\sum_{T=1}^{T_{max}}\frac{1+\eta t - 2\eta}{2}\|\nabla F(\tilde{w}^T)\|^2 \leq& F(w^0) - F(\tilde{w}^T) + \frac{\eta^3 L^2 t(t+1)(2t+1)}{12}TG^2 \\
&+ \frac{m^2 LT}{2K^2}\sum_{k=1}^{K}(\sigma_k^T)^2 + \frac{1}{2}Tc^2\eta^2 t^2 G^2,
\end{aligned} \tag{C.15}$$

And we get

$$\begin{aligned}
\min_{T\in[T_{max}]}\mathbb{E}\|\nabla F(\tilde{w}^T)\|^2 \leq& \frac{2(F(w^0) - F(\tilde{w}^*))}{(1+\eta t - 2\eta)T} + \frac{\eta^3 L^2 t(t+1)(2t+1)G^2}{6(1+\eta t - 2\eta)} \\
&+ \frac{m^2 L\sum_{k=1}^{K}(\sigma_k^T)^2}{K^2(1+\eta t - 2\eta)} + \frac{c^2\eta^2 t^2 G^2}{1+\eta t - 2\eta}.
\end{aligned} \tag{C.16}$$

□

References

[1] Lin J, Yu W, Zhang N, Yang X, Zhang H, Zhao W. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet Things J 2017;4(5):1125–42. http://dx.doi.org/10.1109/JIOT.2017.2683200.

[2] Konečnỳ J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D. Federated learning: Strategies for improving communication efficiency. 2016, arXiv preprint arXiv:1610.05492.

[3] McMahan B, Moore E, Ramage D, Hampson S, Arcas BAy. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th international conference on artificial intelligence and statistics. 54, Sydney, Australia; 2017, p. 1273–82.

[4] Chen M, Mathews R, Ouyang T, Beaufays F. Federated learning of out-of-vocabulary words. 2019, arXiv preprint arXiv:1903.10635.

[5] Yang T, Andrew G, Eichner H, Sun H, Li W, Kong N, et al. Applied federated learning: Improving google keyboard query suggestions. 2018, arXiv preprint arXiv:1812.02903.

[6] Ramaswamy S, Mathews R, Rao K, Beaufays F. Federated learning for emoji prediction in a mobile keyboard. 2019, arXiv preprint arXiv:1906.04329.

[7] Han X, Yu H, Gu H. Visual inspection with federated learning. In: Proceedings of the 2019 international conference on image analysis and recognition. Waterloo, Canada; 2019, p. 52–64.

[8] Xu J, Glicksberg BS, Su C, Walker P, Bian J, Wang F. Federated learning for healthcare informatics. J Healthc Informs Res 2021;5(1):1–19. http://dx.doi.org/10.1007/s41666-020-00082-4.

[9] Brisimi TS, Chen R, Mela T, Olshevsky A, Paschalidis IC, Shi W. Federated learning of predictive models from federated electronic health records. Int J Med Inform 2018;112:59–67. http://dx.doi.org/10.1016/j.ijmedinf.2018.01.007.

[10] Qolomany B, Ahmad K, Al-Fuqaha A, Qadir J. Particle swarm optimized federated learning for industrial IoT and smart city services. In: Proceeding of the 2020 IEEE global communications conference. 2020, p. 1–6.

[11] Dwork C, Naor M. On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. J Priv Confid 2010;2(1). http://dx.doi.org/10.29012/jpc.v2i1.585.

[12] Melis L, Song C, De Cristofaro E, Shmatikov V. Exploiting unintended feature leakage in collaborative learning. In: Proceedings of the 2019 IEEE symposium on security and privacy. San Francisco, USA; 2019, p. 691–706.

[13] Nasr M, Shokri R, Houmansadr A. Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. 2018, arXiv preprint arXiv:1812.00910.

[14] Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. USA: Denver; 2015, p. 1322–33. http://dx.doi.org/10.1145/2810103.2813677.

[15] Zhu L, Liu Z, Han S. Deep leakage from gradients. In: Proceedings of the advances in neural information processing systems. 32, Vancouver, Canada; 2019.

[16] Xu G, Li H, Liu S, Yang K, Lin X. VerifyNet: Secure and verifiable federated learning. IEEE Trans Inf Forensics Secur 2020;15:911–26. http://dx.doi.org/10.1109/TIFS.2019.2929409.

[17] Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans Inf Forensics Secur 2018;13(5):1333–45. http://dx.doi.org/10.1109/TIFS.2017.2787987.

[18] Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. Dallas, USA; 2017, p. 1175–91.

[19] Chen X, Wu SZ, Hong M. Understanding gradient clipping in private SGD: A geometric perspective. In: Advances in neural information processing systems. 33, 2020, p. 13773–82.

[20] Geyer RC, Klein T, Nabi M. Differentially private federated learning: A client level perspective. 2017, arXiv preprint arXiv:1712.07557.

[21] Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. Vienna, Austria; 2016, p. 308–18. http://dx.doi.org/10.1145/2976749.2978318.

[22] Hardy S, Henecka W, Ivey-Law H, Nock R, Patrini G, Smith G, Thorne B. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. 2017, arXiv preprint arXiv:1711.10677.

[23] Chabanne H, de Wargny A, Milgram J, Morel C, Prouff E. Privacy-preserving classification on deep neural network. In: Cryptology ePrint Archive. Report, 2017/035, 2017.

[24] Shamir A. How to share a secret. Commun ACM 1979;22(11):612–3. http://dx.doi.org/10.1145/359168.359176.

[25] Chaum D. The dining cryptographers problem: Unconditional sender and recipient untraceability. J Cryptol 1988;1(1):65–75. http://dx.doi.org/10.1007/BF00206326.

[26] Truex S, Liu L, Gursoy ME, Yu L, Wei W. Demystifying membership inference attacks in machine learning as a service. IEEE Trans Serv Comput 2021;14(6):2073–89. http://dx.doi.org/10.1109/TSC.2019.2897554.

[27] Dwork C, Roth A. The algorithmic foundations of differential privacy. Found Trends in Theor Comput Sci 2014;9(3–4):211–407. http://dx.doi.org/10.1561/0400000042.

[28] Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. Colorado, USA: Denver; 2015, p. 1310–21. http://dx.doi.org/10.1145/2810103.2813687.

[29] Wei K, Li J, Ding M, Ma C, Yang HH, Farokhi F, et al. Federated learning with differential privacy: Algorithms and performance analysis. IEEE Trans Inf Forensics Secur 2020;15:3454–69. http://dx.doi.org/10.1109/TIFS.2020.2988575.

[30] Xiong Z, Cai Z, Takabi D, Li W. Privacy threat and defense for federated learning with non-i.i.d. Data in aIoT. IEEE Trans Ind Inf 2022;18(2):1310–21. http://dx.doi.org/10.1109/TII.2021.3073925.

[31] Naseri M, Hayes J, De Cristofaro E. Local and central differential privacy for robustness and privacy in federated learning. In: Proceedings of the 2022 network and distributed system security symposium. 2022.

[32] Wei K, Li J, Ding M, Ma C, Su H, Zhang B, et al. User-level privacy-preserving federated learning: Analysis and performance optimization. IEEE Trans Mob Comput 2022;21(9):3388–401. http://dx.doi.org/10.1109/TMC.2021.3056991.

[33] Rahman MA, Rahman T, Laganière R, Mohammed N, Wang Y. Membership inference attack against differentially private deep learning model. Trans Data Priv 2018;11(1):61–79.

[34] Hitaj B, Ateniese G, Perez-Cruz F. Deep models under the GAN: Information leakage from collaborative deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017, p. 603–18. http://dx.doi.org/10.1145/3133956.3134012.

[35] Pichapati V, Suresh AT, Yu FX, Reddi SJ, Kumar S. AdaCliP: Adaptive clipping for private SGD. 2019, arXiv preprint arXiv:1908.07643.

[36] Zhang J, He T, Sra S, Jadbabaie A. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In: International conference on learning representations. 2020.

[37] Nissim K, Raskhodnikova S, Smith A. Smooth sensitivity and sampling in private data analysis. In: Proceedings of the 39th annual ACM symposium on theory of computing. San Diego, USA; 2007, p. 75–84. http://dx.doi.org/10.1145/1250790.1250803.

[38] Andrew G, Thakkar O, McMahan HB, Ramaswamy S. Differentially private learning with adaptive clipping. 2019, arXiv preprint arXiv:1905.03871.

[39] Shokri R, Stronati M, Song C, Shmatikov V. Membership inference attacks against machine learning models. In: Proceedings of the 2017 IEEE symposium on security and privacy. San Jose, USA; 2017, p. 3–18. http://dx.doi.org/10.1109/SP.2017.41.

[40] Backes M, Berrang P, Humbert M, Manoharan P. Membership privacy in microrna-based studies. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. Vienna, Austria; 2016, p. 319–30. http://dx.doi.org/10.1145/2976749.2978355.

[41] Homer N, Szelinger S, Redman M, Duggan D, Tembe W, Muehling J, et al. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. PLoS Genet 2008;4(8):e1000167. http://dx.doi.org/10.1371/journal.pgen.1000167.

[42] Pyrgelis A, Troncoso C, De Cristofaro E. Knock knock, who's there? Membership inference on aggregate location data. 2017, arXiv preprint arXiv:1708.06145.

[43] Dwork C, Smith A, Steinke T, Ullman J, Vadhan S. Robust traceability from trace amounts. In: Proceedings of 2015 IEEE 56th annual symposium on foundations of computer science. 2015, p. 650–69. http://dx.doi.org/10.1109/FOCS.2015.46.

[44] Hayes J, Melis L, Danezis G, De Cristofaro E. Logan: Membership inference attacks against generative models. In: Proceedings of the privacy enhancing technologies. Barcelona, Spain; 2019, p. 133–52. http://dx.doi.org/10.2478/popets-2019-0008.

[45] Ateniese G, Mancini LV, Spognardi A, Villani A, Vitali D, Felici G. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. Int J Secur Netw 2015;10(3):137–50. http://dx.doi.org/10.1504/IJSN.2015.071829.

[46] Ganju K, Wang Q, Yang W, Gunter CA, Borisov N. Property inference attacks on fully connected neural networks using permutation invariant representations. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. Toronto, Canada; 2018, p. 619–33. http://dx.doi.org/10.1145/3243734.3243834.

[47] Liu T, Hu X, Shu T. Assisting backdoor federated learning with global knowledge alignment. 2021, https://drive.google.com/file/d/1L3694k1GXGnByfcREZUE0P-gwjzGkU3T/view?usp=sharing.

[48] Wang L, Xu S, Wang X, Zhu Q. Eavesdrop the composition proportion of training labels in federated learning. 2019, arXiv preprint arXiv:1910.06044.

[49] Kargupta H, Datta S, Wang Q, Sivakumar K. On the privacy preserving properties of random data perturbation techniques. In: Proceedings of the 3rd IEEE international conference on data mining. 2003, p. 99–106. http://dx.doi.org/10.1109/ICDM.2003.1250908.

[50] Liu D, Simeone O. Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control. IEEE J Sel Areas Commun 2021;39(1):170–85. http://dx.doi.org/10.1109/JSAC.2020.3036948.
[51] Minka T. Estimating a Dirichlet distribution. Technical report, MIT; 2000.
[52] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on computer vision and pattern recognition. 2016, p. 770–8.
[53] Krizhevsky A, Hinton G, et al. Learning multiple layers of features from tiny images. Citeseer; 2009.
[54] Amiri MM, Gündüz D, Kulkarni SR, Vincent Poor H. Convergence of federated learning over a noisy downlink. IEEE Trans Wireless Commun 2021;1. http://dx.doi.org/10.1109/TWC.2021.3103874.