Federated Learning with Sparsification-Amplified Privacy and Adaptive Optimization

Rui Hu , **Yanmin Gong*** and **Yuanxiong Guo**
The University of Texas at San Antonio
{rui.hu, yanmin.gong, yuanxiong.guo}@utsa.edu

Abstract

Federated learning (FL) enables distributed agents to collaboratively learn a centralized model without sharing their raw data with each other. However, data locality does not provide sufficient privacy protection, and it is desirable to facilitate FL with rigorous differential privacy (DP) guarantee. Existing DP mechanisms would introduce random noise with magnitude proportional to the model size, which can be quite large in deep neural networks. In this paper, we propose a new FL framework with sparsification-amplified privacy. Our approach integrates random sparsification with gradient perturbation on each agent to amplify privacy guarantee. Since sparsification would increase the number of communication rounds required to achieve a certain target accuracy, which is unfavorable for DP guarantee, we further introduce acceleration techniques to help reduce the privacy cost. We rigorously analyze the convergence of our approach and utilize Renyi DP to tightly account the end-to-end DP guarantee. Extensive experiments on benchmark datasets validate that our approach outperforms previous differentially-private FL approaches in both privacy guarantee and communication efficiency.

1 Introduction

Federated learning (FL) is a new distributed learning paradigm that enables multiple agents to collaboratively learn a shared model under the orchestration of the cloud without sharing their local data [McMahan *et al.*, 2017]. By keeping data locally, FL is advantageous in privacy and communication efficiency compared with traditional centralized learning paradigm. However, recent inference attacks [Fredrikson *et al.*, 2015; Shokri *et al.*, 2017] show that the local model updates shared between agents could also lead to privacy leakage, and it is desirable to protect the shared local model updates with rigorous privacy guarantee[1].

To address this issue, several privacy-preserving framework have been proposed, among which differential privacy (DP) [Dwork and Roth, 2014] has become the de-facto standard due to its rigorous privacy guarantee and effectiveness in data analysis tasks [Abadi *et al.*, 2016; Hu *et al.*, 2020; Huang *et al.*, 2019; Guo and Gong, 2018; Gong *et al.*, 2016]. General DP mechanisms, such as Gaussian or Laplacian mechanism, rely on the injection of carefully calibrated noise to the output of an algorithm directly. This poses new challenges to achieving DP in FL because the added noise is proportional to the model size which can be very large with modern deep learning neural networks (e.g., millions of model parameter), resulting in significantly degraded model accuracy. Under the local DP setting where the cloud is not fully trusted, the challenges become more prominent as all the local updates shared with the cloud need to be protected.

Existing works on differentially-private machine learning either consider centralized DP [Abadi *et al.*, 2016], or rely on costly techniques such as secure multi-party computation [Truex *et al.*, 2019] and shuffling via anonymous channels [Liu *et al.*, 2020; Erlingsson *et al.*, 2019] to remove the requirement of a trusted cloud and improve the model accuracy in local DP. How to better balance the model accuracy and privacy protection in FL efficiently remains largely unknown.

In this paper, we propose a novel differentially-private FL scheme, called Fed-SPA, to provide strong privacy guarantee in the local DP setting while maintaining high model accuracy. In light of the observation that local model updates in FL are largely sparse, we design a sparsification-coded DP mechanism that integrates gradient perturbation with random sparsification to amplify the privacy guarantee with little sacrifice on the model accuracy. Random sparsification transforms a large vector into a sparse one by keeping only a random subset of coordinates while setting other coordinates to zeros. As we will show in this paper, random sparsification not only introduces randomness to the scheme, but also reduces the sensitivity of the shared model updates with respect to raw data, thus resulting in smaller privacy loss at every communication round. Furthermore, as sparsification can slow down the convergence speed of learning algorithms and increase the total number of communication rounds, we propose to further reduce the end-to-end privacy loss by using convergence acceleration techniques to offset the negative impact of sparsification. We provide theoretical analysis to demonstrate the

*Contact Author

[1]The full version is available at https://arxiv.org/abs/2008.01558

convergence of our scheme and rigorous privacy guarantee.

The main contributions of this paper are summarized below.

- We propose to use sparsification for privacy amplification in FL while also improving communication efficiency. Previous works that consider both communication efficiency and DP treat them as two separate goals and solve them in an uncoordinated manner. Unlike previous approaches, we focus on the interplay of those two goals and aim to kill two birds with one stone in this paper, i.e., use sparsification as a tool to improve DP and achieve communication efficiency at the same time. We theoretically analyze the impacts of sparsification on the utility-privacy trade-off and design a sparsification-coded DP mechanism for FL that provides stronger privacy guarantee with the same amount of random noise.
- To further improve the utility-privacy trade-off, we integrate the sparsification-coded DP mechanisms with convergence acceleration techniques, which can reduce the number of required communication rounds and ensure faster convergence. Specifically, we adapt an acceleration strategy similar to that of Adam optimizer to the FL setting and design an adaptive aggregation strategy on the cloud to reduce the number of communication rounds. The resulting scheme called Fed-SPA provides a better utility-privacy trade-off than the state-of-art differentially private FL methods.
- We empirically evaluate our scheme on several benchmark datasets. The experiment results show that Fed-SPA significantly boosts the model accuracy, and at the same time saves more than 80% of the bandwidth cost compared to the state-of-art approaches under the same DP guarantee.

It is worth noting that our scheme aims to improve the utility-privacy trade-off while achieving communication efficiency. This distinguishes our paper from previous studies on communication efficient and differentially-private distributed learning that focus on ensuring privacy protection and achieving communication efficiency at the same time. For example, cpSGD [Agarwal *et al.*, 2018] is a modified distributed SGD scheme which is private and communication-efficient via gradient quantization and binomial mechanism. However, since quantization does not provide any privacy amplification effects as sparsification, the utility-privacy trade-off is not improved in that approach.

2 Preliminaries

DP is a rigorous notion of privacy that has become the defacto standard for measuring privacy risk. In the context of FL, DP ensures that the exchanged model updates are nearly the same regardless of the usage of a data sample. In this paper, we consider a relaxed DP definition called Rényi differential privacy (RDP), which is strictly stronger than (ϵ, δ)-DP for $\delta > 0$ and allows tighter composition analysis.

Definition 1 ((α, ρ)-RDP)**.** *Given a real number $\alpha \in (1, +\infty)$ and privacy parameter $\rho \geq 0$, a randomized mechanism $\mathcal{M}$ satisfies (α, ρ)-RDP if for any two neighboring datasets D, D' that differs in one record, the Rényi α-divergence between $\mathcal{M}(D)$ and $\mathcal{M}(D')$ satisfies*

$$D_\alpha[\mathcal{M}(D)\|\mathcal{M}(D')] := \frac{1}{\alpha - 1} \log \mathbb{E}\left[\left(\frac{\mathcal{M}(D)}{\mathcal{M}(D')}\right)^\alpha\right] \leq \rho,$$

where the expectation is taken over the output of $\mathcal{M}(D')$.

Lemma 1 (RDP Composition [Mironov, 2017])**.** *If $\mathcal{M}_1$ satisfies (α, ρ_1)-RDP and $\mathcal{M}_2$ satisfies (α, ρ_2)-RDP, then their composition $\mathcal{M}_1 \circ \mathcal{M}_2$ satisfies $(\alpha, \rho_1 + \rho_2)$-RDP.*

Lemma 2 (Gaussian Mechanism [Mironov, 2017])**.** *Let $h : \mathcal{D} \to \mathbb{R}^d$ be a vector-valued function over datasets. The Gaussian mechanism $\mathcal{M} = h(D) + \mathbf{b}$ with $\mathbf{b} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$ satisfies $(\alpha, \alpha\phi^2(h)/2\sigma^2)$-RDP, where $\phi(h)$ is the L_2 sensitivity of h defined by $\phi(h) = \sup_{D,D'} \|h(D) - h(D')\|_2$ with D, D' being two neighboring datasets in $\mathcal{D}$.*

3 Fed-SPA: Federated Learning with Sparsification-Amplified Privacy and Adaptive Optimization

Notation. We use $[n]$ to denote the set of integers $\{1, 2, \ldots, n\}$ with any positive integer n, and $[\cdot]_j$ to denote the j-th coordinate of a vector. Let $\|\cdot\|$ be the ℓ_2 vector norm.

3.1 Problem Formulation

A typical FL system consists of n agents and a central server (e.g., the cloud). Each agent $i \in [n]$ has a local dataset with m data samples, and all agents collaboratively train a global model $\boldsymbol{\theta}$ on the collection of their local datasets under the orchestration of the central server. The agents in FL aim to find the optimal global model $\boldsymbol{\theta}$ by solving the following empirical risk minimization problem while keeping their data locally:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} f(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta}), \tag{1}$$

where $f_i(\boldsymbol{\theta}) = \mathbb{E}_{z \sim \mathcal{D}_i}[l_i(\boldsymbol{\theta}; z)]$ represents the loss function of i-th agent (possibly non-convex), $\mathcal{D}_i$ is the data distribution of i-th agent, and z represents a data sampled from $\mathcal{D}_i$. For $i \neq j$, the data distributions $\mathcal{D}_i$ and $\mathcal{D}_j$ may be very different.

Threat Model. Before elaborating the proposed solutions, we first define the following threat model considered in this paper. The adversary considered here can be the "honest-but-curious" aggregation server or agents in the system. The aggregation server will honestly follow the designed training protocol but are curious about agents' private data and may infer it from the shared messages. Furthermore, some agents can collude with the aggregation server or each other to infer private information about a specific victim agent. Besides, the adversary could also be the passive outside attacker. These attackers can eavesdrop all shared messages in the execution of the training protocol but will not actively inject false messages into or interrupt message transmissions.

3.2 Classic FL Algorithm: Federated Averaging

As the most widely-used algorithm in the FL setting, Federated Averaging (FedAvg) [McMahan *et al.*, 2017] solves (1) by selecting and distributing the current global model to a subset of agents, running multiple steps of SGD in parallel on the selected agents, and then aggregating the model updates from those agents to improve the global model iteratively. Specifically, FedAvg involves T communication rounds, and each round consists of four stages: First, at the beginning of round $t \in \{0, \ldots, T-1\}$, the server selects a subset of agents $\mathcal{W} \subseteq [n]$ to participate and sends them the latest global model $\boldsymbol{\theta}_t$. Second, each agent $i \in \mathcal{W}$ initializes its local model $\boldsymbol{\theta}_i^{t,0}$ to be the global model $\boldsymbol{\theta}_t$ and then performs τ iterations of SGD on its local dataset as follows:

$$\boldsymbol{\theta}_i^{t,s+1} = \boldsymbol{\theta}_i^{t,s} - \eta_l \boldsymbol{g}_i^{t,s}, \quad s = 0, 1, \ldots, \tau - 1 \tag{2}$$

where η_l is the local learning rate. Here, $\boldsymbol{g}_i^{t,s} := (1/B) \sum_{z \in \xi_i^{t,s}} \nabla l(\boldsymbol{\theta}_i^{t,s}, z)$ represents the stochastic gradient computed on a mini-batch $\xi_i^{t,s}$ of B samples, which a unbiased estimate of $\nabla f_i(\boldsymbol{\theta}_i^{t,s})$. Third, each agent $i \in \mathcal{W}$ uploads the final local model update $\boldsymbol{\theta}_i^{t,\tau} - \boldsymbol{\theta}_t$ to the server. Fourth, the server aggregates the local model updates from all participating agents and improves the global model as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \frac{1}{|\mathcal{W}|} \sum_{i \in \mathcal{W}} (\boldsymbol{\theta}_i^{t,\tau} - \boldsymbol{\theta}_t). \tag{3}$$

The same procedure repeats for the next round.

Privacy and Communication Drawbacks. Although FedAvg avoids the direct information leakage by keeping data locally, the intermediate updates exchanged during the collaboration process such as $\boldsymbol{\theta}_i^{t,\tau} - \boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_t$ could still leak private information about the local data as demonstrated in recent advanced attacks such as model inversion attacks [Fredrikson *et al.*, 2015] and membership attacks [Shokri *et al.*, 2017]. Furthermore, in FedAvg, agents need to repeatedly upload local model updates (i.e., $\boldsymbol{\theta}_t - \boldsymbol{\theta}_i^{t,\tau}$) of large size (e.g., millions of model parameters for modern deep neural network models) to the server and download the newly-updated global model (i.e., $\boldsymbol{\theta}_t$) from the server in order to learn an accurate global model (e.g., ~ 1000 rounds for running CNN on MINIST dataset or ~ 4000 for LSTM on Shakespeare dataset to reach 99% accuracy [McMahan *et al.*, 2017]). Since the privacy loss is proportional to the model dimension and the number of communication rounds, the size of added DP noise could be very large in order to provide a strong local DP guarantee, which will degrade the model accuracy heavily. Besides, as the bandwidth between the server and agents could be rather limited in practice (e.g., wireless connection between the cloud and smartphones), especially for uplink transmissions, the overall communication cost could be very high. The above drawbacks motivate us to develop a new privacy-preserving and communication-efficient FL scheme.

3.3 Proposed Fed-SPA Algorithm

In this subsection, we present Fed-SPA, our proposed FL scheme with the goal of improving user privacy protection and also communication efficiency while maintaining high model accuracy. To ensure easy integration into existing packages/systems, Fed-SPA follows the same overall structure of FedAvg but differs in the following two key aspects: 1) local models are updated at each agent using a variant of SGD, where the gradients are perturbed by our proposed sparsification-coded DP mechanism; and 2) the global model is updated adaptively instead of simple averaging to accelerate convergence at the server. The entire process of Fed-SPA is summarized in Algorithm 1.

User-Side Sparsification-Coded DP Mechanism. To address privacy and communication aspects simultaneously, we design a sparsification-coded DP mechanism, which integrates Gaussian mechanism and sparsification to reduce the privacy loss of each local iteration and the size of transmitted local model updates at each communication round. Specifically, we use the rand_k sparsifier to reduce the message size by a factor of d/k, which is defined as follows:

Definition 2 (rand_k Sparsifier)**.** *For parameter $k \in [d]$, the operator $\mathrm{rand}_k : \mathbb{R}^d \times \Omega_k \to \mathbb{R}^d$ is defined for a vector $\mathbf{x} \in \mathbb{R}^d$ as*

$$[\mathrm{rand}_k(\mathbf{x}, \omega)]_j := \begin{cases} [\mathbf{x}]_j, & \text{if } j \in \omega, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $\Omega_k = \binom{[d]}{k}$ denotes the set of all k-element subsets of $[d]$. We omit the second argument whenever it is chosen uniformly at random, i.e., $\omega \sim_{u.a.r} \Omega_k$.

With the rand_k sparsifier, our sparsification-coded DP mechanism works as follows: at communication round t, each selected agent $i \in \mathcal{W}$ first generates its own sparsifier $\mathrm{rand}_k(\cdot)$ by randomly sampling a set of k active coordinates ω_i^t before performing τ SGD updates, and then, at s-th local iteration, perturbs and sparsifies the stochastic gradient using Gaussian noise and the generated sparsifier $\mathrm{rand}_k(\cdot)$. We use the same sparsifier (with active set ω_i^t) across all local iterations of communication round t on agent i so that the transmitted model update $(\boldsymbol{\theta}_i^{t,\tau} - \boldsymbol{\theta}_t)$ is still a sparse vector (which has active coordinates ω_i^t), preserving the communication efficiency benefit of sparsification. Let $p = k/d$ represent the compression ratio, the update rule at each agent is:

$$\boldsymbol{\theta}_i^{t,s+1} = \boldsymbol{\theta}_i^{t,s} - \eta_l S_i^t(\boldsymbol{g}_i^{t,s} + \boldsymbol{b}_i^{t,s}), \tag{5}$$

where $S_i^t(\cdot) := (1/p)\,\mathrm{rand}_k(\cdot)$ is a scaled variant of $\mathrm{rand}_k(\cdot)$, and $\boldsymbol{b}_i^{t,s}$ is the noise sampled from the Gaussian distribution $\mathcal{N}(0, \sigma^2 \mathbf{I}_d)$. We use the scaled sparsifier $S_i^t(\cdot)$ so that the sparsified noisy gradient is an unbiased estimate of the true gradient.

Server-Side Adaptive Update. The sparsification-coded DP mechanism will inevitably slow down the convergence speed due to the increased variance of the stochastic gradient used in each iteration, and the privacy loss of each agent increases proportionally with the number of iterations according to the composition property of DP in Lemma 1. To improve the privacy, we speed up the convergence by updating the model in an adaptive manner similar as Adam [Kingma and Ba, 2014] on the server. The adaptive optimizer like

Algorithm 1 The Fed-SPA Algorithm

Require: Initial model $\boldsymbol{\theta}_0$, initial momentums $[\boldsymbol{v}_{-1}]_j \geq \kappa^2, \forall j \in [d]$, $\boldsymbol{u}_{-1} = \mathbf{0}_d$, noise magnitude σ, number of rounds T, number of local iterations τ, compression ratio p, momentum parameters β_1, β_2, learning rates η_l, η_g, and batch size B.

1: **for** $t = 0$ to $T-1$ **do**
2: Randomly selects a set of agents $\mathcal{W}$
3: Broadcasts $\boldsymbol{\theta}_t$ to all agents in $\mathcal{W}$
4: **for** each agent $i \in \mathcal{W}$ in parallel **do**
5: Generate a new sparsifier $S_i^t(\cdot)$
6: $\boldsymbol{\theta}_i^{t,0} \leftarrow \boldsymbol{\theta}_t$
7: **for** $s = 0$ to $\tau - 1$ **do**
8: Compute a stochastic gradient $\boldsymbol{g}_i^{t,s}$ over a mini-batch $\xi_i^{t,s}$ of B samples
9: $\boldsymbol{\theta}_i^{t,s+1} \leftarrow \boldsymbol{\theta}_i^{t,s} - \eta_l S_i^t(\boldsymbol{g}_i^{t,s} + \boldsymbol{b}_i^{t,s})$ where $\boldsymbol{b}_i^{t,s} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$
10: **end for**
11: $\Delta_i^t \leftarrow \boldsymbol{\theta}_i^{t,\tau} - \boldsymbol{\theta}_t$ and upload Δ_i^t to the server
12: **end for**
13: $\boldsymbol{u}_t \leftarrow \beta_1 \boldsymbol{u}_{t-1} + (1-\beta_1) \sum_{i \in \mathcal{W}} \Delta_i^t / |\mathcal{W}|$
14: $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2) \boldsymbol{u}_t^2$
15: $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \eta_g \boldsymbol{u}_t / (\sqrt{\boldsymbol{v}_t} + \kappa)$
16: **end for**

Adam can be modified by adding noise to their gradients to provide better DP guarantee, in terms of reducing the iterations needed to achieve a target model accuracy [Yu *et al.*, 2018]. However, there are two main constraints unique to deploying the adaptive optimizer on each agent in the FL setting. First, it is often the case that each agent participate only once or several times intermittently during the entire training process, and hence, replacing SGD with an adaptive optimizer at each agent during the local update stage will perform poorly due to the stale historical information such as the momentum in Adam. Second, maintaining the historical information on resource-constrained agents (e.g. smartphones) is costly for computation and storage resource. To address the above issues, in Fed-SPA, the server, rather than the agents, will carry out the adaptive update without any additional communication. The server maintains two momentum vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$ which get updated at each round. Specifically, at round t, after τ local iterations, each agent $i \in \mathcal{W}$ uploads its model update $\Delta_i^t := \boldsymbol{\theta}_i^{t,\tau} - \boldsymbol{\theta}_t$ to the server to improve the global model as follows:

$$\begin{cases} \boldsymbol{u}_t = \beta_1 \boldsymbol{u}_{t-1} + (1-\beta_1) \sum_{i \in \mathcal{W}} \Delta_i^t / |\mathcal{W}|, \\ \boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2) \boldsymbol{u}_t^2, \\ \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta_g \boldsymbol{u}_t / (\sqrt{\boldsymbol{v}_t} + \kappa), \end{cases} \tag{6}$$

where $\beta_1, \beta_2 \in [0, 1)$ are momentum parameters, η_g is the global learning rate, and κ controls the degree of adaptivity. Note that the math operations in (6) are element-wise.

4 Main Theoretical Results

In this section, we give the formal privacy guarantee and rigorous convergence analysis of Fed-SPA. Before stating our results, we make the following assumptions:

Assumption 1 (Smoothness)**.** *The local objective function f_i is L-smooth, i.e., for any $i \in [n]$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we have $f_i(\mathbf{y}) \leq f_i(\mathbf{x}) + \langle \nabla f_i(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + (L/2)\|\mathbf{y} - \mathbf{x}\|^2$.*

Assumption 2 (Bounded Variance)**.** *Let $\boldsymbol{g}_i$ be the stochastic gradient over the mini-batch sampled from the distribution $\mathcal{D}_i$. The function f_i has a bounded local variance, i.e., $\mathbb{E}\|[\boldsymbol{g}_i]_j - [\nabla f_i(\mathbf{x})]_j\|^2 \leq \zeta_{l,j}^2$ for all $\mathbf{x} \in \mathbb{R}^d$, $j \in [d]$ and $i \in [n]$. Moreover, the global variance is bounded, i.e., $(1/n)\sum_{i=1}^n \mathbb{E}\|[\nabla f_i(\mathbf{x})]_j - [\nabla f(\mathbf{x})]_j\|^2 \leq \zeta_{g,j}^2$ for all $\mathbf{x} \in \mathbb{R}^d$, $j \in [d]$ and $i \in [n]$. We also denote $\zeta_l^2 := \sum_{j=1}^d \zeta_{l,j}^2$ and $\zeta_g^2 := \sum_{j=1}^d \zeta_{g,j}^2$ for convenience.*

Assumption 3 (Bounded Gradient)**.** *The loss function $l_i(\mathbf{x}, z)$ has $G/\sqrt{d}$-bounded gradients, i.e., for any data sample z from $\mathcal{D}_i$, we have $|[\nabla l_i(\mathbf{x}, z)]_j| \leq G/\sqrt{d}$ for all $\mathbf{x} \in \mathbb{R}^d$, $j \in [d]$ and $i \in [n]$.*

Assumption 1 is standard and implies that the global loss function f is also L-smooth. Assumption 2 and Assumption 3 are fairly standard in non-convex optimization literature [Reddi *et al.*, 2020; Kingma and Ba, 2014]. Assumption 3 characterizes the sensitivity of each coordinate of gradient $\nabla l(\mathbf{x}, z)$ and implies $\mathbb{E}\|\nabla l(\mathbf{x}, z)\|^2 \leq G^2$, which can be enforced by the gradient clipping technique.

4.1 Privacy Analysis

In this subsection, we discuss how the sparsification in our sparsification-coded DP mechanism amplifies the privacy and provide the end-to-end privacy guarantee of Fed-SPA.

In our sparsification-coded DP mechanism, the rand_k sparsifier does not provide any DP guarantee by itself, but it amplifies the privacy guarantee provided by additive Gaussian noises. To analyze the end-to-end privacy, we first need to analyze the sensitivity of the sparsified gradient in Line 9, Algorithm 1. With the rand_k sparsifier, the active coordinate set ω is chosen independent of data and does not leak privacy, and only the values of active coordinates $\{[\mathbf{x}]_j, j \in \omega\}$ contain private data information and need to be protected. Therefore, in our sparsification-coded DP mechanism, only values of active coordinates of the gradient are actually perturbed by the Gaussian noise. More precisely, let ω_i^t denote the active coordinate set for participating agent i at round t, the sparsified noisy gradient at each local iteration can be represented as $S_i^t(\boldsymbol{g}_i^{t,s} + \boldsymbol{b}_i^{t,s}) = [\boldsymbol{g}_i^{t,s} + \boldsymbol{b}_i^{t,s}]_{\omega_i^t}/p = [\boldsymbol{g}_i^{t,s}]_{\omega_i^t}/p + [\boldsymbol{b}_i^{t,s}]_{\omega_i^t}/p$, where we can observe that the amount of added noise is proportional to k, the number of active coordinates, and is reduced by a factor of d/k compared with the standard Gaussian mechanism. In the following, we analyze the sensitivity of $[\boldsymbol{g}_i^{t,s}]_{\omega_i^t}$ and then compute the privacy guarantee after adding noise $[\boldsymbol{b}_i^{t,s}]_{\omega_i^t}$. For agent i, given any two neighboring datasets $\xi_i^{t,s}$ and $\grave{\xi}_i^{t,s}$ that have the same size B but differ in one data sample (e.g., $z \in \xi_i^{t,s}$ and $\grave{z} \in \grave{\xi}_i^{t,s}$). The L_2 sensitivity of $[\boldsymbol{g}_i^{t,s}]_{\omega_i^t}$ can be denoted as $\phi_{\omega_i^t}^2 = \max \|[\boldsymbol{g}_i^{t,s}]_{\omega_i^t} - [\nabla f_i(\boldsymbol{\theta}_i^{t,s}, \grave{\xi}_i^{t,s})]_{\omega_i^t}\|^2 = \max \|(1/B)[\nabla l(\boldsymbol{\theta}_i^{t,s}, z) - \nabla l(\boldsymbol{\theta}_i^{t,s}, \grave{z})]_{\omega_i^t}\|^2$ and is upper-

bounded by $2pG^2/B^2$ following Assumption 3. We observe that the sensitivity of $[\boldsymbol{g}_i^{t,s}]_{\omega_i^t}$ is proportional to the compression ratio p, reducing the privacy loss according to Lemma 2.

Then, we give the end-to-end DP guarantee of Fed-SPA in Theorem 1 and provide the proof in Appendix E. Given a fixed value of δ, ϵ is computed numerically by searching an optimal α that minimizes ϵ. We observe that the noise magnitude σ is proportional to p, which implies that sparsification, i.e., when $p < 1$, can reduce the magnitude of Gaussian noise and hence improve the model accuracy.

Theorem 1 (Privacy Guarantee)**.** *Suppose the mini-batch $\xi_i^{t,s}$ is sampled without replacement at each iteration. Let $q := B/m$ be the data sampling rate. Let I_i represent the number of rounds agent i participates during the training. Under Assumption 3, if $\sigma'^2 = \sigma^2 B^2/2pG^2 \geq 0.7$, Fed-SPA achieves (ϵ, δ)-DP for agent i, where*

$$\epsilon = \frac{7q^2 I_i \tau \alpha p G^2}{B^2 \sigma^2} + \frac{\log(1/\delta)}{\alpha - 1},$$

for any $\alpha \leq (2/3)\sigma^2 \log\left(1/q\alpha(1+\sigma'^2)\right) + 1$ and $\delta \in (0,1)$.

4.2 Convergence Analysis

In this subsection, we present the convergence results of Fed-SPA for general loss functions satisfying Assumptions 1-3. For ease of illustration, we assume *full participation*, i.e., $|\mathcal{W}| = n$, and $\beta_1 = 0$, though our analysis can be easily extended to $\beta_1 > 0$ and *partial participation* (i.e., $|\mathcal{W}| < n$, see Appendix H for details). As $f(\cdot)$ may be non-convex, we study the gradient of the global model $\boldsymbol{\theta}_t$ as t increases. We give the result in Theorem 2 and the proof in the full version.

Theorem 2 (Convergence Result)**.** *Let Assumptions 1-3 hold, and L, G, ζ_l, ζ_g be as defined therein. Suppose the local learning rate satisfies $\eta_l \leq \min\{1/8L\tau, (1/8\tau)\min\{\kappa\sqrt{d}/G, (\kappa^2\sqrt{d}/G\eta_g L)^{1/2}\}\}$. Let $\zeta_{dp}^2 = (G^2 + \zeta_l^2)/p + pd\sigma^2$, then the iterates of Algorithm 1 satisfy:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{\theta}_t)\|^2 = \mathcal{O}\left((\sqrt{\beta_2}\eta_l\tau G/\sqrt{d} + \kappa)(\Xi + \Xi')\right)$$

with

$$\Xi = \frac{f(\boldsymbol{\theta}_0) - f^*}{\eta_l \eta_g \tau T} + \frac{5\eta_l^2 \tau L^2}{2\kappa}\left(\zeta_{dp}^2 + 6\tau\zeta_g^2\right),$$

$$\Xi' = \left(\frac{\eta_g L}{2} + \frac{G}{\sqrt{d}}\right)\left[\frac{4\eta_l}{n\kappa^2}\zeta_{dp}^2 + \frac{20\eta_l^3\tau^2 L^2}{\kappa^2}\left(\zeta_{dp}^2 + 6\tau\zeta_g^2\right)\right],$$

where f^* is the optimal objective value.

We restate the above result for a specific choice of η_l, η_g and κ in Lemma 3 to highlight the dependence on τ and T. Note that when T is sufficiently large compared to τ, $\mathcal{O}(1/\sqrt{n\tau T})$ is the dominant term in Lemma 3. Therefore, Fed-SPA converges at a rate of $\mathcal{O}(1/\sqrt{n\tau T})$, which matches the best known rate for the general non-convex setting of our interest [Reddi *et al.*, 2020]. We also note that both sparsification and Gaussian noise will slow down the convergence. However, a small compression ratio p can reduce the amount of added noise (i.e., the term $pd\sigma^2$) by a factor of $1/p$, which implies the privacy amplification effect of sparsification.

Lemma 3. *Choose the local learning rate $\eta_l = \Theta(1/L\tau\sqrt{T})$ that satisfies the condition in Theorem 2. Suppose $\eta_g = \Theta(\sqrt{n\tau})$ and $\kappa = G/\sqrt{d}L$, then the iterates of Algorithm 1 satisfy*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{\theta}_t)\|^2 = \mathcal{O}\left(\frac{f(\boldsymbol{\theta}_0) - f^*}{\sqrt{n\tau T}} + \frac{2\zeta_{dp}^2 L}{G^2\sqrt{n\tau T}} + \frac{(\zeta_{dp}^2 + 6\tau\zeta_g^2)}{G\tau T} + \frac{(\zeta_{dp}^2 + 6\tau\zeta_g^2)L\sqrt{n}}{G^2\sqrt{\tau}T^{3/2}}\right),$$

when T is sufficiently large.

5 Experimental Evaluation

The goal of this section is to evaluate the performance of Fed-SPA on several benchmark datasets. We aim at evaluating the performance of Fed-SPA with different levels of compression and comparing them with the performance of the following three schemes: 1) FedAvg: the classic FL algorithm; 2) DP-Fed: this baseline follows the algorithm of FedAvg except that the stochastic gradient is perturbed by adding Gaussian noise $\mathcal{N}(0, \sigma^2\mathbf{I}_d)$; 3) cpSGD [Agarwal *et al.*, 2018]: this baseline follows the algorithm of classic distributed SGD except that the stochastic gradient is quantized into some discrete domain and then perturbed using Binomial mechanism.

5.1 Experimental Setup

We explore two widely-used benchmark datasets in FL: MNIST [LeCun *et al.*, 1998] and CIFAR-10 [Krizhevsky *et al.*, 2009]. The MNIST dataset consists of 10 classes of 28×28 handwritten digit images. There are 60K training examples and 10K testing examples, which are partitioned among 100 agents, each containing 600 training and 100 testing examples. The CIFAR-10 dataset consists of 10 classes of 32×32 images. There are 50K training examples and 10K testing examples in the dataset, which are partitioned into 100 agents, each containing 500 training and 100 testing examples. We use a CNN model for the MNIST dataset, which has two 5×5 convolutional layers (the first with 10 filters, the second with 20 filters, each followed with 2×2 max pooling and ReLu activation), a fully connected layer with 50 units and ReLu activation, and a final softmax output layer. For the CIFAR-10 dataset, we use a CNN model that consists of three 3×3 convolution layers (the first with 64 filters, the second with 128 filters, the third with 256 filters, each followed with 2×2 max pooling and ReLu activation), two fully connected layers (the first with 128 units, the second with 256 units, each followed with ReLu activation), and a final softmax output layer.

We set the privacy failure probability $\delta = 10^{-3}$ and the sampling ratio of agents $r = |\mathcal{W}|/n = 0.1$ for all experiments by default. Since cpSGD follows the classic distributed SGD scheme, we have $\tau = 1$ and $r = 1.0$ for cpSGD. We tune the hyperparameters using grid-search. We set the number of local iterations $\tau = 300$ for MNIST and $\tau = 50$ for CIFAR-10. The details of other hyperparameter settings are given in the full version. The per-coordinate sensitivity $G/\sqrt{d}$ is selected during an initialization round for each scheme by taking the median value over N absolute values.

Compression ratio	Algorithm	Performance		
		Accuracy (%)	Cost (MB)	ϵ
$p = 0.05$	Fed-SPA	92.65	0.0197	1.0
	cpSGD	diverge	-	-
$p = 0.1$	Fed-SPA	92.16	0.0393	1.0
	cpSGD	diverge	-	-
$p = 0.4$	Fed-SPA	94.84	0.1572	1.0
	cpSGD	87.32	1.5720	3.63
$p = 1.0$	Fed-SPA	94.70	0.3931	1.0
	cpSGD	88.48	3.9310	2.26
	DP-Fed	91.41	0.3931	1.0
	FedAvg	96.87	0.3931	-

Table 1: Summary of results on MNIST dataset.

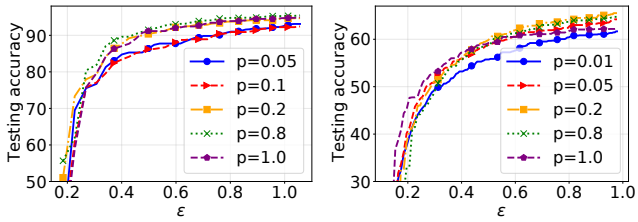
5.2 Experimental Results

Table 1 shows the best accuracy over 45 communication rounds for each scheme on the MNIST dataset, and Table 2 represents the best accuracy over 200 rounds for each scheme on the CIFAR-10 dataset. Assume each value of the model parameter is represented by a 32-bit floating number. The *Compression ratio* for Fed-SPA is calculated as $p = k/d$. For cpSGD, $p = \log_2(m + b)/32$, where b implies the b-bit quantization and m is the parameter for the Binomial distribution $\text{Bin}(m, 0.5)$. For FedAvg and DP-Fed, we have $p = 1.0$. *Cost* is the average bandwidth consumption calculated as $p \times d \times 32 \times T \times r$, where r is the sampling probability of devices and T is the communication rounds. Note that since how to analyze Binomial mechanism in cpSGD using RDP is still unknown, we account the privacy loss ϵ of cpSGD using the standard subsampling amplification [Balle *et al.*, 2018] and advanced composition [Dwork *et al.*, 2010].

Without compression, i.e., when $p = 1.0$, Fed-SPA outperforms cpSGD and DP-Fed on both datasets as the server in Fed-SPA updates the global model adaptively to speed up the convergence, and its best accuracy to achieve $(1.0, 10^{-3})$-DP is close to the best accuracy of FedAvg under the same communication cost. Note that since the Binomial noise used in cpSGD is not as concentrated as the Gaussian noise used in Fed-SPA and DP-Fed, the model accuracy of cpSGD degrades heavily, and hence to achieve a target accuracy, the privacy loss of cpSGD is very large. As the compression ratio p decreases, the performance of Fed-SPA on both datasets does not degrade much, compared with cpSGD which cannot

Compression ratio	Algorithm	Performance		
		Accuracy (%)	Cost (MB)	ϵ
$p = 0.05$	Fed-SPA	63.0	-	1.0
	cpSGD	diverge	-	-
$p = 0.1$	Fed-SPA	63.28	2.15	1.0
	cpSGD	diverge	-	-
$p = 0.4$	Fed-SPA	64.36	8.60	1.0
	cpSGD	35.74	86.02	258
$p = 1.0$	Fed-SPA	62.06	21.50	1.0
	cpSGD	37.17	215.04	39.77
	DP-Fed	61.13	21.50	1.0
	FedAvg	67.16	21.50	-

Table 2: Summary of results on CIFAR-10 dataset.



(a) MNIST. (b) CIFAR-10.

Figure 1: Privacy-accuracy trade-off of Fed-SPA.

achieve a reasonable privacy guarantee when $p \leq 0.1$. More importantly, Fed-SPA for MNIST achieves a higher accuracy than DP-Fed under the same privacy cost, while saving 86% communication cost when $p = 0.05$. The reason is that decreasing p also lowers the sensitivity of the sparsified gradient which has a direct impact on the noise magnitude to achieve a target privacy guarantee, as explain in Section 4.1. We observe a similar trend for CIFAR-10 from Table 2: Fed-SPA performs better when the compression ratio is small, e.g., when $p \leq 0.4$, as the sensitivity in this case is small.

Then, we show the privacy-accuracy trade-off of Fed-SPA for both datasets with different levels of compression in Figure 1. As we mentioned above, a small compression ratio results in a smaller sensitivity, and hence less Gaussian noise will be added to the model. On the other hand, the compression ratio cannot be arbitrarily small as it will increase the variance of gradient as explained in Section 4.2. Therefore, we should find an optimal p that is small enough to reduce the size of additive Gaussian noise but large enough to avoid large compression error. For both datasets, we can observe that when the privacy budget ϵ is limited (e.g., $\epsilon < 1.0$), a small enough p (e.g., $p = 0.8$ for MNIST and $p = 0.2$ for CIFAR-10) achieves higher accuracy as it reduces the size of Gaussian noise. As ϵ increases, the size of Gaussian noise decreases, and hence Fed-SPA with larger p performs better due to the smaller compression error.

6 Conclusion

This paper has proposed Fed-SPA, a new FL scheme based on sparsification-coded DP mechanism and server-side adaptive update, to improve privacy-accuracy trade-off and achieve communication efficiency at the same time. We have provided rigorous convergence and privacy analysis of Fed-SPA. Extensive experiments based on benchmark datasets have been conducted to verify the effectiveness of the proposed scheme and numerically show the trade-off between privacy guarantee and model accuracy. For future work, we plan to investigate the interplay of privacy protection with other communication-efficient techniques in FL.

Acknowledgements

The work of R. Hu and Y. Gong was supported by the U.S. National Science Foundation under grants US CNS-2029685 and CNS-1850523. The work of Y. Guo was supported by the U.S. National Science Foundation under grant US CNS-2029685.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[Agarwal *et al.*, 2018] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*, pages 7564–7575, 2018.

[Balle *et al.*, 2018] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems*, pages 6277–6287, 2018.

[Dwork and Roth, 2014] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[Dwork *et al.*, 2010] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[Erlingsson *et al.*, 2019] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM, 2019.

[Fredrikson *et al.*, 2015] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[Gong *et al.*, 2016] Yanmin Gong, Yuguang Fang, and Yuanxiong Guo. Private data analytics on biomedical sensing data via distributed computation. *IEEE/ACM transactions on computational biology and bioinformatics*, 13(3):431–444, 2016.

[Guo and Gong, 2018] Yuanxiong Guo and Yanmin Gong. Practical collaborative learning for crowdsensing in the internet of things with differential privacy. In *2018 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2018.

[Hu *et al.*, 2020] Rui Hu, Yuanxiong Guo, Hongning Li, Qingqi Pei, and Yanmin Gong. Personalized federated learning with differential privacy. *IEEE Internet of Things Journal*, 2020.

[Huang *et al.*, 2019] Zonghao Huang, Rui Hu, Yuanxiong Guo, Eric Chan-Tin, and Yanmin Gong. DP-ADMM: ADMM-based distributed learning with differential privacy. *IEEE Transactions on Information Forensics and Security*, 15:1002–1012, 2019.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Liu *et al.*, 2020] Ruixuan Liu, Yang Cao, Hong Chen, Ruoyang Guo, and Masatoshi Yoshikawa. FLAME: Differentially private federated learning in the shuffle model. *arXiv preprint arXiv:2009.08063*, 2020.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[Mironov, 2017] Ilya Mironov. Renyi differential privacy. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*, pages 263–275. IEEE, 2017.

[Reddi *et al.*, 2020] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[Shokri *et al.*, 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[Truex *et al.*, 2019] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11, 2019.

[Wang *et al.*, 2019a] Lingxiao Wang, Bargav Jayaraman, David Evans, and Quanquan Gu. Efficient privacy-preserving nonconvex optimization. *arXiv preprint arXiv:1910.13659*, 2019.

[Wang *et al.*, 2019b] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019.

[Yu *et al.*, 2018] Da Yu, Huishuai Zhang, and Wei Chen. Improve the gradient perturbation approach for differentially private optimization. *NeurIPS 2018 Workshop*, 2018.