

User-Level Privacy-Preserving Federated Learning: Analysis and Performance Optimization

Kang Wei[✉], *Student Member, IEEE*, Jun Li[✉], *Senior Member, IEEE*,
Ming Ding[✉], *Senior Member, IEEE*, Chuan Ma[✉], *Member, IEEE*, Hang Su, *Member, IEEE*,
Bo Zhang, *Senior Member, IEEE*, and H. Vincent Poor[✉], *Life Fellow, IEEE*

Abstract—Federated learning (FL), as a type of collaborative machine learning framework, is capable of preserving private data from mobile terminals (MTs) while training the data into useful models. Nevertheless, from a viewpoint of information theory, it is still possible for a curious server to infer private information from the shared models uploaded by MTs. To address this problem, we first make use of the concept of local differential privacy (LDP), and propose a user-level differential privacy (UDP) algorithm by adding artificial noise to the shared models before uploading them to servers. According to our analysis, the UDP framework can realize (ϵ_i, δ_i) -LDP for the i th MT with adjustable privacy protection levels by varying the variances of the artificial noise processes. We then derive a theoretical convergence upper-bound for the UDP algorithm. It reveals that there exists an optimal number of communication rounds to achieve the best learning performance. More importantly, we propose a communication rounds discounting (CRD) method. Compared with the heuristic search method, the proposed CRD method can achieve a much better trade-off between the computational complexity of searching and the convergence performance. Extensive experiments indicate that our UDP algorithm using the proposed CRD method can effectively improve both the training efficiency and model quality for the given privacy protection levels.

Index Terms—Federated learning, differential privacy, communication round, mobile edge computing

1 INTRODUCTION

WITH the dramatic development of the internet-of-things (IoT), the amount of data originating from intelligent devices is growing at unprecedented rates [1], [2]. Conventional machine learning (ML) is no longer capable of efficiently processing such data in a centralized manner. To address this challenge, several distributed ML architectures have been proposed with different approaches of aggregating gradients or models [3]. However, data privacy and confidentiality [4], [5], [6] are of concern in such approaches as exchanged gradients or models usually contain clients' sensitive information. One such distributed ML architecture is federated learning (FL), which allows a decoupling of data provision at end-user equipment (UE) and machine learning model aggregation at a central server [7], [8]. In FL, all mobile terminals (MTs) with the same data structure collaboratively learn a shared model with the help of a server, i.e.,

training the model at MTs and aggregating model parameters at the server. Owing to the local training, FL does not require MTs to upload their private data, thereby effectively reducing transmission overhead as well as promoting MTs' privacy. As such, FL is applicable to a variety of scenarios where data are either sensitive or expensive to be transmitted to the server, e.g., health-care records, private images, and personally identifiable information, etc. [9], [10], [11].

Although FL can help prevent private data from being exposed to the public, hidden adversaries may attack the learning model by eavesdropping and analyzing the shared parameters, e.g., via a reconstruction attack [12] or an inference attack [13]. For example, a malicious classifier may reveal the features of the MTs' data and reconstruct data points from the FL training process [12]. Some designed attack strategies can be found in recent studies. The work in [14] recovered the private data based on the observation that the gradient of weights is proportional to that of the bias, and their ratio approximates the training input. The work in [12] considered an untrusted server in FL and proposed a generative adversarial network (GAN) based reconstruction attack. Furthermore, this reconstruction attack utilized the shared model as the discriminator to train a GAN [15] model that generates original samples of the training data. In [13], Melis *et al.* demonstrated that the shared models in FL may leak unintended information from participants' training data, and they developed passive and active inference attacks to exploit this leakage. The work in [16] showed that it is possible to obtain the private training data from the publicly shared gradients, i.e., deep

- Kang Wei, Jun Li, and Chuan Ma are with the School of Electrical and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China. E-mail: {kang.wei, jun.li, chuan.ma}@njust.edu.cn.
- Ming Ding is with the Data61, CSIRO, Sydney, ACT 2601, Australia. E-mail: ming.ding@data61.csiro.au.
- Hang Su and Bo Zhang are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: {suhangss, dcszb}@tsinghua.edu.cn.
- H. Vincent Poor is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA. E-mail: poor@princeton.edu.

Manuscript received 10 July 2020; revised 23 Jan. 2021; accepted 26 Jan. 2021.
Date of publication 4 Feb. 2021; date of current version 4 Aug. 2022.
(Corresponding author: Jun Li.)
Digital Object Identifier no. 10.1109/TMC.2021.3056991

leakage from gradient, which was empirically validated on both computer vision and natural language processing tasks. The work in [12] proposed a novel GAN framework with a multi-task discriminator at the server side to attack user-level privacy, which can simultaneously discriminate category, reality, and user identity of input samples. Therefore, it is challenging to preserve data contributors' privacy from [17].

Therefore, privacy-preserving ML has attracted intensive attention in recent years, as the emergence of centralized searchable data repositories and open data applications may lead to the leakage of private information. The work in [3] first proposed the concept of deep learning with DP, providing an evaluation criterion for privacy guarantees. The work in [18] improved DP based stochastic gradient descent (SGD) algorithms by carefully allocation of a privacy budget at each training iteration. In [18], the privacy budget and step size for each iteration are dynamically determined at runtime based on the quality of the noisy statistics (e.g., gradient) obtained for the current training iteration. Privacy issues are also more critical in distributed ML. The work in [19] introduced the notion of DP in distributed ML and proposed a distributed online learning algorithm to improve the learning performance for a given privacy level. The work in [20] analyzed the privacy loss in a DP-based distributed ML framework, and provided the explicit convergence performance. The work in [21] presented a theoretical analysis of DP-based SGD algorithms, and provided an approach for analyzing the quality of ML models related to the privacy level and the size of the datasets.

A formal treatment of privacy risks in FL calls for a holistic and interdisciplinary approach [22]. The work in [23] proposed an FL algorithm based on a random sub-sampling scheduling at each aggregation. This algorithm can achieve good training performance at a given privacy level when there are a sufficiently large number of participating MTs. However, it cannot preserve MTs' private information from being exposed to a curious server. The work in [24] presented an alternative approach that utilizes both DP and secure multiparty computation (SMC) to protect against inference threats and produces models with high accuracy. A key component of this work is the ability to reduce noise by leveraging the SMC framework while considering a customizable trust parameter, which will also consume more communication and computing resources. The work in [25] involved sketching algorithms, using hash functions to compress the input data with bounded errors, to consider communication efficiency and privacy protection in distributed learning. However, some of these approaches cannot preserve MTs' private information from being exposed to the curious server. Moreover, other approaches, such as SMC and sketching algorithms, can consume large amounts of communications and computer resources.

To solve these challenges, it is important to design an effective algorithm to mitigate the privacy concerns for data sharing without deteriorating the quality of trained FL models. In this paper, in order to prevent information leakage from the shared model parameters and improve the training efficiency, we develop a novel privacy-preserving FL framework, termed the user-level differential privacy (UDP) algorithm. Furthermore, we develop a theoretical convergence bound for this UDP algorithm and design a novel online

TABLE 1
Summary of Main Notation

\mathcal{M}	A randomized mechanism for LDP
x, x'	Adjacent databases
ϵ, δ	The parameters related to the original LDP
ϵ_i, δ_i	The parameters related to LDP for the i th MT
\mathcal{C}_i	The i th MT
\mathcal{D}_i	The dataset held by the owner \mathcal{C}_i
$\mathcal{D}_{i,m}$	The m th sample in \mathcal{D}_i
\mathcal{D}	The dataset held by all the MTs
$ \cdot $	The cardinality of a set
\mathcal{U}	The set of all users
U	Total number of all users ($U = \mathcal{U} $)
\mathcal{K}	The set of chosen MTs
K	The number of chosen MTs ($K = \mathcal{K} $)
t	The index of the t th communication round
T	The number of communication rounds
w	The vector of model parameters
$F(w)$	Global loss function
$F_i(w)$	Local loss function from the i th MT
w_i^t	Local training parameters of the i th MT
\tilde{w}_i^t	Local training parameters after adding noises
w^0	Initial parameters of the global model
w^t	Global parameters generated from local parameters at the t th communication round
w^*	The optimal parameters that minimize $F(w)$
C	The clipping threshold C
σ_i	The STD of artificial noises for LDP
η	The learning rate of SGD
g_i^t	The local gradient of i th MT
$g_{i,m}^t$	The local gradient of m th sample held by i th MT $\alpha(\lambda)$
$\ell(\cdot)$	The moment generating function λ Any positive integer
$\Delta\ell$	The local training process
ν_0	The sensitivity of the local training process
ν_1	The Gaussian probability density function of $\mathcal{N}(0, \sigma_i^2)$
μ	The mixture of two Gaussian distributions
L	$q\mathcal{N}(\Delta\ell, \sigma_i^2) + (1-q)\mathcal{N}(0, \sigma_i^2)$
ϵ_i	The coefficient of Polyak-Lojasiewicz condition
ζ	The constant determined by the practical loss function
β	The divergence metric of i th MT
$\mathcal{V}(w^t)$	The threshold to determine when to trigger the discounting method
	The linear discounting factor
	The test loss by the model w^t

optimization method to achieve better convergence performance compared with the original UDP algorithm.

Specifically, the contributions of this paper can be summarized as follows:

- We introduce a novel privacy-preserving FL framework, namely the user-level differential privacy algorithm, using the concept of local differential privacy (LDP) and verify that this UDP framework can realize (ϵ_i, δ_i) -LDP for the i th MT for iterative learning model exchange with a curious server based on our analysis.
- We enhance a standard differential privacy (DP) mechanism (the moments accountant method) and analyze the sensitivity for each MT under the LDP definition. Then, we prove that the training process of the i th MT satisfies the requirement of (ϵ_i, δ_i) -LDP for different privacy levels by properly adapting their variances of Gaussian noise added to the model updates.

- We also show that there exists an optimal number of communication rounds in terms of convergence performance for a given privacy level. This property demonstrates that we need to have a new look at communication rounds. Thus, we design an online optimization method, termed communication rounds discounting (CRD), which can obtain a better tradeoff between complexity and convergence performance compared with the original UDP algorithm and an offline heuristic search method.
- We conduct extensive experiments on real-world datasets and the experimental results validate that our CRD method in UDP can achieve performance equivalent to that of offline search in terms of loss function values, but with a much lower complexity.

The remainder of this paper is organized as follows. The threat model and background on LDP and FL are presented in Section 2. Then, we provide the details of the proposed UDP algorithm and the privacy analysis in Section 3, and present the noise recalculation and CRD methods in Section 4. The analytical and experimental results are shown in Section 5. Finally, conclusions are drawn in Section 6.

2 PRELIMINARIES

In this section, we first present the FL framework and the threat model, and then introduce the basic knowledge of (ϵ, δ) -LDP.

2.1 Federated Learning

Let us consider a general FL system consisting of a centralized server and U MTs. Let \mathcal{D}_i denote the local dataset held by MT C_i , where $\mathcal{U} = \{1, 2, \dots, U\}$ and $i \in \mathcal{U}$. For all participants, the objective is to learn a global model over data that resides at the U associated MTs. Formally, this FL task can be expressed as

$$w^* = \arg \min_w F(\mathcal{D}, w), \quad (1)$$

where $F(\mathcal{D}, w) = \sum_{i \in \mathcal{U}} p_i F_i(\mathcal{D}_i, w)$, $F_i(\cdot)$ is the local loss function of the i th MT, $p_i = |\mathcal{D}_i|/|\mathcal{D}| \geq 0$ with $\sum_{i \in \mathcal{U}} p_i = 1$, $|\mathcal{D}_i|$ is the number of data samples in the i th MT's dataset and $|\mathcal{D}| = \sum_{i \in \mathcal{U}} |\mathcal{D}_i|$ is the total number of data samples in all MTs' datasets, respectively. Generally, the local loss functions $F_i(\cdot)$ are given by local empirical risks and have the same expression for various MTs. At the server, K MTs are chosen and a model aggregation is performed over their uploaded models. In particular, w is the global model parameter, given by

$$w = \sum_{i \in \mathcal{K}} p_i w_i, \quad (2)$$

where w_i is the parameter vector trained at the i th MT, \mathcal{K} is a subset of \mathcal{U} , with K MTs out of U MTs chosen for participating in the model aggregation, respectively. To facilitate analysis on the privacy performance, in this paper, we adopt a low complexity method, termed K -user random scheduling, in which K MTs are randomly chosen from all the MTs.

2.2 Threat Model

In this paper, we assume that the server is honest-but-curious, which means it will strictly follow the FL rule, but may

recover the training datasets [12] or infer private features [13] based on the local uploaded parameters. Concretely, this curious server can train a GAN framework, e.g., multi-task GAN-AI [12], which may simultaneously discriminates category, reality, and user identity of input samples. This novel discrimination on user identity enables the generator to recover users' private data. In addition, this curious server may also be interested in learning whether a given sample belongs to the training datasets, which can be inferred by utilizing the difference between model outputs from training and non-training this sample [26]. For example, when FL is conducting a clinical experiment, a participant may not want the observer to know whether he is involved in this experiment (a MT may contain several individuals' records). However, the adversary may link the test results to the appearance or disappearance of this participant, and then possibly inflict harm to this person. Therefore, our threat model is reasonable and realistic.

2.3 Local Differential Privacy

DP mechanism with parameters ϵ and δ provides a strong criterion for the privacy preservation of distributed data processing systems. Here, $\epsilon > 0$ is the distinguishable bound of all outputs on neighboring datasets x, x' in a database \mathcal{X} , and δ represents the probability of the event that the ratio of the probabilities for two adjacent datasets x, x' cannot be bounded by e^ϵ after adding a privacy-preserving mechanism. With an arbitrarily given δ , a larger ϵ gives a clearer distinguishability of neighboring datasets and hence a higher risk of privacy violation. Now, we will formally define LDP as follows.

Definition 1. ((ϵ, δ) -LDP [27]): A randomized mechanism \mathcal{M} satisfies (ϵ, δ) -DP: $\mathcal{X} \rightarrow \mathcal{R}$ with domain \mathcal{X} and range \mathcal{R} , for all measurable sets $\mathcal{S} \subseteq \mathcal{R}$ and any two adjacent datasets $x, x' \in \mathcal{X}$, we have

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

In this paper, we choose the Gaussian mechanism that adopts L_2 norm sensitivity. It adds zero-mean Gaussian noise with variance $\sigma^2 \mathbf{I}$ in each coordinate of the function output $s(x)$ as

$$\mathcal{M}(x) = s(x) + \mathcal{N}(0, \sigma^2 \mathbf{I}), \quad (4)$$

where \mathbf{I} is an identity matrix and has the same size with $s(x)$. The sensitivity of the function s can be expressed as

$$\Delta s = \max_{x, x' \in \mathcal{X}} \|s(x) - s(x')\|_2, \quad (5)$$

which gives an upper bound on how much we must perturb its output considering preserving privacy. It satisfies (ϵ, δ) -LDP when we properly select the value of σ .

3 PRIVACY ANALYSIS

In this section, we first propose a user-level differential privacy algorithm in FL against the curious server. Then, we develop a method for analyzing the privacy loss moment by improving the conventional analysis approach [3]. Based on our improved method, we are able to accurately calculate the standard deviation (STD) of additive noises in UDP to guarantee (ϵ_i, δ_i) -LDP for the i th MT.

3.1 User-Level DP

In this subsection, to prevent information leakage from uploaded parameters, we will introduce a UDP algorithm, which borrows the concept of LDP. Our UDP algorithm is outlined in Algorithm 1 for training an effective model with the (ϵ_i, δ_i) -LDP guarantee for the i th MT. We can note that all MTs have their own privacy parameters ϵ_i and δ_i . We denote by T the number of communication rounds, by w^0 the initial global parameter, by σ_i the STD of additive noises for the i th MT, by t the index of the current communication round and by q the random sampling ratio ($q = K/U$).

Algorithm 1. User-Level DP (UDP)

Data: The number of communication rounds T , the initial global parameter w^0 , the sample ratio $q = K/U$, the clipping threshold C and the LDP parameters (ϵ_i, δ_i) for all MTs

- 1: Initialize: $t = 0$
- 2: The server broadcasts w^0 and T to all MTs
- 3: **while** $t < T$ **do**
- 4: **for** $i \in \mathcal{K}$ **do**
- 5: Update the local gradients:
- 6: $g_{i,m}^t(\mathcal{D}_{i,m}) = \nabla_{w^t} F_i(\mathcal{D}_{i,m}, w^t)$, where $m \in \{1, \dots, |\mathcal{D}_i|\}$;
- 7: Clip the local gradients:
- 8: $\tilde{g}_{i,m}^t(\mathcal{D}_{i,m}) = g_{i,m}^t(\mathcal{D}_{i,m}) / \max(1, \frac{\|g_{i,m}^t(\mathcal{D}_{i,m})\|_2}{C})$;
- 9: Update the local parameters:
- 10: $w_i^{t+1} = w^t - \frac{1}{|\mathcal{D}_i|} \sum_{m=1}^{|\mathcal{D}_i|} \eta \tilde{g}_{i,m}^t(\mathcal{D}_{i,m})$;
- 11: Calculate σ_i according to LDP parameters (ϵ_i, δ_i) :
- 12: $\tilde{w}_i^{t+1} = w_i^{t+1} + \mathcal{N}(0, \sigma_i^2 \mathbf{I})$;
- 13: upload noised parameters to the server;
- 14: Update the global parameters w^{t+1} as
- 15: $w^{t+1} = \sum_{i \in \mathcal{K}} p_i \tilde{w}_i^{t+1}$;
- 16: The server broadcasts the global parameters
- 17: **for** $C_i \in \{C_1, C_2, \dots, C_U\}$ **do**
- 18: Test the aggregating parameters w^{t+1} ;
- 19: using local dataset
- 20: $t = t + 1$;

Result: w^T

At the beginning, the server broadcasts the initiate global parameter w^0 and T to all MTs. Then, K active MTs are chosen and train the model parameters by using local datasets with preset terminal conditions, respectively. During the local training, the local gradients $g_{i,m}^t, \forall m \in \{1, \dots, |\mathcal{D}_i|\}$ are clipped by the threshold C . After the local training, the i th MT, $\forall i \in \mathcal{K}$, will add noises to the trained model parameters w_i^{t+1} , in which σ_i^2 is the variance of artificial noises and this value is calculated by the i th MT according to the privacy level (ϵ_i, δ_i) , sampling ratio q and the number of communication rounds T . When all active MTs finish local training process, they are required to upload the noised parameters \tilde{w}_i^{t+1} to the server for aggregation.

Then, the server updates the global parameters w^{t+1} by aggregating the local parameters integrated with different weights according to (2) and broadcasts them to the MTs. The accuracy of each MT will be estimated based on the received global parameters w^{t+1} using local testing datasets. If preset terminal conditions are not satisfied, all MTs will start the next round of training process based on these updated global parameters. In detail, when the aggregation time reaches a preset number of communication rounds T , our UDP completes and returns w^T . Owing to the local perturbations, it will be difficult for the honest-but-curious

server to infer private features of the i th-MT. In this case, in order to effectively protect the i th MT privacy, the STD of additive noises will be analyzed according to the concept of (ϵ_i, δ_i) -LDP in the following sections.

3.2 Bound of the Moment

Before calculating the STD σ_i of the additive noises, we will first enhance the classic loss moment analysis method in [3]. According to [3], using Gaussian mechanism, we can define privacy loss of i th MT after T communication rounds in our UDP algorithm by

$$c \triangleq \exp(\alpha^T(\lambda)) = \exp\left(\sum_{t=1}^T \alpha(\lambda)\right), \quad (6)$$

where the moment generating function $\alpha(\lambda)$ is given by

$$\alpha(\lambda) \triangleq \ln(\max\{D_{v_1, v_0}, D_{v_0, v_1}\}), \quad (7)$$

λ is any positive integer, v_0 denotes the Gaussian probability density function (PDF) of $\mathcal{N}(0, \sigma_i^2)$, v_1 denotes the mixture of two Gaussian distributions $q\mathcal{N}(\Delta\ell, \sigma_i^2) + (1-q)\mathcal{N}(0, \sigma_i^2)$, $q = K/U$ is the random sampling ratio in UDP algorithm (means that all MTs have the same probability to be selected in the aggregation) and $\Delta\ell$ denotes the sensitivity of the local training process ℓ , respectively. The expression of D_{v_1, v_0} and D_{v_0, v_1} can be written as

$$D_{v_1, v_0} = \mathbb{E}_{z \sim v_1} \left(\frac{v_1}{v_0} \right)^\lambda = \mathbb{E}_{z \sim v_0} \left(\frac{v_1}{v_0} \right)^{\lambda+1}, \quad (8)$$

and

$$D_{v_0, v_1} = \mathbb{E}_{z \sim v_0} \left(\frac{v_0}{v_1} \right)^\lambda = \mathbb{E}_{z \sim v_0} \left(\frac{v_1}{v_0} \right)^{-\lambda}. \quad (9)$$

However, the derivations in [3] for the bound of moments can only be applied for the rigorous constraint $q \leq \frac{1}{16\sigma_i^2}$. To tackle this problem, we propose Lemma 1 to further bound the moment.

Lemma 1. Considering two Gaussian distributions v_0 and v_1 used in the moments accountant method, they satisfy the following relationship:

$$D_{v_1, v_0} \geq D_{v_0, v_1}. \quad (10)$$

Proof. See Appendix A, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TMC.2021.3056991>. \square

Note that the only difference between the D_{v_1, v_0} and D_{v_0, v_1} is the factor of $\lambda + 1$ and $-\lambda$ on the exponent. With Lemma 1, we can directly obtain that $\alpha(\lambda) = D_{v_1, v_0}$. In this way, we can calculate the privacy loss by bound D_{v_1, v_0} , which can relax the constraint $q \leq \frac{1}{16\sigma_i^2}$. In the following subsection, we will derive the STD of the Gaussian noises added in the UDP algorithm with Lemma 1.

3.3 User-Level Noise Calculation

With the sensitivity and the bound of the moment, we can design the Gaussian mechanism $\mathcal{N}(0, \sigma_i)$ for the i th MT with (ϵ_i, δ_i) -LDP requirement in terms of the sampling ratio

$q = K/U$ and the number of communication rounds T . The STD of the Gaussian noises can be derived according to the following theorem.

Theorem 1. Given the sampling ratio q and the number of communication rounds T , to guarantee (ϵ_i, δ_i) -LDP for the i th MT, the STD of noises σ_i from Gaussian mechanism should satisfy

$$\sigma_i = \frac{\Delta \ell \sqrt{2qT \ln(1/\delta_i)}}{\epsilon_i}, \quad (11)$$

where $\Delta \ell$ is the sensitivity of the local training process and $\ell(\cdot)$ denotes the local training process.

Proof. In this proof, we first need to calculate the sensitivity of the local training process defined by (5). According to the definition of LDP, we consider two adjacent datasets $\mathcal{D}_i, \mathcal{D}'_i \in \mathcal{X}$ in the i th MT, where \mathcal{D}_i and \mathcal{D}'_i have the same size, and only differ by one sample. Consequently, for the i th MT with the training dataset \mathcal{D}_i , the t th local training process can be written into the following form:

$$\mathbf{w}_i^{t+1} = \ell(\mathcal{D}_i, \mathbf{w}^t), \quad (12)$$

Therefore, the sensitivity of the local training process can be given as

$$\Delta \ell = \max_{\mathcal{D}_i, \mathcal{D}'_i \in \mathcal{X}} \|\ell(\mathcal{D}_i, \mathbf{w}^t) - \ell(\mathcal{D}'_i, \mathbf{w}^t)\|. \quad (13)$$

Assuming that the batch size in the local training is equal to the number of training samples, we have

$$\begin{aligned} \Delta \ell &= \frac{\eta}{|\mathcal{D}_i|} \sum_{m=1}^{|\mathcal{D}_i|} \max_{\mathcal{D}_{i,m}, \mathcal{D}'_{i,m} \in \mathcal{X}} \left\| \mathbf{g}_{i,m}^t(\mathcal{D}_{i,m}) \right. \\ &\quad \left. - \mathbf{g}_{i,m}^t(\mathcal{D}'_{i,m}) \right\| \leq \frac{2\eta C}{|\mathcal{D}_i|}, \end{aligned} \quad (14)$$

where C is the clipping threshold to bound $\|\mathbf{g}_{i,m}^t\|$.

Then, we need to calculate the privacy loss using the moments account method [3]. Hence, the λ th moment $\alpha(\lambda)$ can be expressed as (7). Here, we want to bound D_{v_0, v_1} and D_{v_1, v_0} . Based on Lemma 1, we only need to bound D_{v_1, v_0} and have

$$\begin{aligned} D_{v_1, v_0} &= \mathbb{E}_{z \sim v_1} \left(\frac{v_1}{v_0} \right)^\lambda \\ &= \int_{-\infty}^{+\infty} v_0 \left(1 - q + qe^{\frac{2z\Delta\ell - \Delta\ell^2}{2\sigma_i^2}} \right)^{\lambda+1} dz \\ &= \int_{-\infty}^{+\infty} v_0 \sum_{l=0}^{\lambda+1} \binom{\lambda+1}{l} (1-q)^{\lambda+1-l} q^l e^{\frac{l(2z\Delta\ell - \Delta\ell^2)}{2\sigma_i^2}} dz \\ &= \sum_{l=0}^{\lambda+1} \binom{\lambda+1}{l} (1-q)^{\lambda+1-l} q^l e^{\frac{l(l-1)\Delta\ell^2}{2\sigma_i^2}} \\ &\leq \left(1 - q + qe^{\frac{\Delta\ell^2}{2\sigma_i^2}} \right)^{\lambda+1} \leq e^{q(\lambda+1) \left(e^{\frac{\Delta\ell^2}{2\sigma_i^2}} - 1 \right)}. \end{aligned} \quad (15)$$

Assuming $\frac{\lambda\Delta\ell^2}{2\sigma_i^2} \ll 1$ for $\lambda \in [1, T]$, we have

$$D_{v_1, v_0} \leq e^{q(\lambda+1) \left(\frac{\Delta\ell^2}{2\sigma_i^2} + O\left(\frac{\lambda^2\Delta\ell^4}{4\sigma_i^4}\right) \right)} \approx e^{\frac{q\lambda(\lambda+1)\Delta\ell^2}{2\sigma_i^2}}. \quad (16)$$

We use the above moments and have

$$\alpha^T(\lambda) \leq \sum_{t=1}^T \alpha(\lambda, \sigma_i) = \frac{Tq\lambda(\lambda+1)\Delta\ell^2}{2\sigma_i^2}. \quad (17)$$

Using the tail bound by moments [3], we have

$$\delta_i = \min_{\lambda} \exp\left(\frac{Tq\lambda(\lambda+1)\Delta\ell^2}{2\sigma_i^2} - \lambda\epsilon_i\right). \quad (18)$$

Considering inequation (18), we know

$$\begin{aligned} \frac{Tq\lambda(\lambda+1)\Delta\ell^2}{2\sigma_i^2} - \lambda\epsilon_i &= \frac{Tq\Delta\ell^2}{2\sigma_i^2} \left(\lambda + \frac{1}{2} - \frac{\epsilon_i\sigma_i^2}{Tq\Delta\ell^2} \right)^2 \\ &\quad - \frac{Tq\Delta\ell^2}{2\sigma_i^2} \left(\frac{1}{2} - \frac{\epsilon_i\sigma_i^2}{Tq\Delta\ell^2} \right)^2. \end{aligned} \quad (19)$$

When setting $\lambda = \frac{\epsilon_i\sigma_i^2}{Tq\Delta\ell^2} - \frac{1}{2}$, we have

$$\begin{aligned} \ln\left(\frac{1}{\delta_i}\right) &\leq \frac{Tq\Delta\ell^2}{2\sigma_i^2} \left(\frac{1}{2} - \frac{\epsilon_i\sigma_i^2}{Tq\Delta\ell^2} \right)^2 \\ &= \frac{Tq\Delta\ell^2}{8\sigma_i^2} - \frac{\epsilon_i}{2} + \frac{\epsilon_i^2\sigma_i^2}{2Tq\Delta\ell^2}. \end{aligned} \quad (20)$$

Since $\delta_i \in (0, 1)$, we can obtain

$$\frac{Tq\lambda(\lambda+1)\Delta\ell^2}{2\sigma_i^2} - \lambda\epsilon_i < 0. \quad (21)$$

Combine inequation (21), we can bound $\ln(1/\delta)$ as

$$\ln\left(\frac{1}{\delta_i}\right) < -\frac{\epsilon_i}{4} + \frac{\epsilon_i^2\sigma_i^2}{2Tq\Delta\ell^2} < \frac{\epsilon_i^2\sigma_i^2}{2Tq\Delta\ell^2}. \quad (22)$$

Therefore, we can choose σ_i satisfies (11) to guarantee (ϵ_i, δ_i) -DP in the UDP framework. \square

Theorem 1 quantifies the relation between the noise level σ_i and the privacy level ϵ_i . It shows that for a fixed perturbation σ_i on model parameters, a larger q leads to a weaker privacy guarantee (i.e., a larger \sqrt{q}/ϵ_i). This is indeed true since when more MTs are involved in computing \mathbf{w} at each communication round, there will be a larger probability of information leakage for each MT. Also, for a given ϵ_i , a larger T in the total training process lead to a higher chance of information leakage because the observer may obtain more information for training datasets. Furthermore, for a given privacy protection level ϵ_i and T , a larger value of q leads to a larger value of σ_i , which helps reduce the i th MT concerns on participating in FL because the i th MT are allowed to add more noise to the trained local models. This requires us to choose the parameters carefully in order to have a reasonable privacy level.

4 DISCOUNTING METHOD IN UDP

In this section, we first introduce the property that there exists an optimal number of communication rounds in the UDP algorithm. Based on this property, we propose a CRD method to improve convergence performance in the training process. Then, we provide a noise calculation method for obtaining the STD of the additive noises in the case of varying T during the FL training. Finally, we will summarize our proposed CRD algorithm.

4.1 Performance Analysis for UDP

First, we start with the essential assumption of on the global loss function $F(\cdot)$ defined by $F(\cdot) \triangleq \sum_{i=1}^U p_i F_i(\cdot)$, and the i th local loss function $F_i(\cdot)$ for the analysis, which can be satisfied normally.

Assumption 1. We assume the following conditions for the loss function of all MTs:

- 1) $F_i(w)$ is convex;
- 2) $F_i(w)$ satisfies the Polyak-Lojasiewicz condition with the positive parameter μ , which implies that $F(w) - F(w^*) \leq \frac{1}{2\mu} \|\nabla F(w)\|^2$, where w^* is the optimal result;
- 3) $F_i(w)$ is L -Lipschitz smooth, i.e., for any w, w' , $\|\nabla F_i(w) - \nabla F_i(w')\| \leq L\|w - w'\|$, where L is a constant determined by the practical loss function;
- 4) $\eta \leq \frac{1}{L}$, where η is the learning rate;
- 5) For any i and w , $\|\nabla F_i(w) - \nabla F(w)\| \leq \epsilon_i$ and $\mathbb{E}\{\epsilon_i\} = \epsilon$, where ϵ_i is the divergence metric.

Based on the above assumptions and Theorem 1, we can obtain the following result which characterizes the convergence performance of the UDP algorithm.

Theorem 2. To guarantee (ϵ_i, δ_i) -DP for all MTs, the convergence upper bound of the UDP algorithm after T communication rounds is given by

$$\mathbb{E}\{F(w^T)\} - F(w^*) \leq A^T (F(w^0) - F(w^*)) + (1 - A^T) \left(\frac{\kappa_0 TK}{U^2} \sum_{i=1}^U \frac{\ln(1/\delta_i)}{\epsilon_i^2} + \frac{\kappa_1 U(U - K)}{K(U - 1)} \right), \quad (23)$$

where $A = 1 - 2\mu\eta + \mu\eta^2 L$, $\kappa_0 = \frac{L^2 \Delta^2}{\mu}$ and $\kappa_1 = \frac{\eta^2 L^2 \epsilon}{2\mu}$.

Proof. See Appendix B, available in the online supplemental material. \square

From Theorem 2, we can find an explicit tradeoff between convergence performance and privacy: When privacy guarantee is weak (large values of ϵ_i and δ_i , and small values of $\sum_{i=1}^U \frac{\ln(1/\delta_i)}{\epsilon_i^2}$), the convergence bound will be small, which indicates a tight convergence to the optimal weights. From our assumptions, we know that $\eta L \leq 1$ and obtain $A < 1$. Then, we show the process on discovering the relationship between the convergence upper bound and the number of total MTs U , the number of participant MTs K and the number of communication rounds T . We find that the number of communication rounds T is a key factor, and then obtain the following theorem.

Theorem 3. There exists an optimal number of communication rounds to achieve the best learning performance for the given $\epsilon_i, \forall i \in \mathcal{U}$, and a sufficiently large U .

Proof. With a slight abuse of notation, we consider continuous values of $U \geq 1$, $1 \leq K \leq U$ and $T \geq 1$. Let $h(U, K, T)$ denote the right hand side (RHS) of (23) and we have

$$\frac{\partial^2 h(U, K, T, \epsilon_i)}{\partial T^2} = A^T \ln A \left(F(w^0) - F(w^*) - \frac{\kappa_0 TK}{U^2} \sum_{i=1}^U \frac{\ln(1/\delta_i)}{\epsilon_i^2} - \frac{\kappa_1 (U - K)}{K(U - 1)} - \frac{\kappa_0 TK}{U \ln A} \sum_{i=1}^U \frac{\ln(1/\delta_i)}{\epsilon_i^2} \right). \quad (24)$$

It can be seen that the first term and fourth term of on the RHS of (24) are always positive. When U and K are set to be large enough, and $\sum_{i=1}^U \frac{\ln(1/\delta_i)}{\epsilon_i^2}$ is small (proper privacy guarantee), we can see that the second term of on the RHS of (24) is small. In this case, we have $\frac{\partial^2 h(U, T, \epsilon_i)}{\partial T^2} > 0$ and the upper bound is convex for T .

Then we consider the condition $K = U$, we have

$$\frac{\partial^2 h(U, T, \epsilon_i)}{\partial T^2} = A^T \ln A \left(F(w^0) - F(w^*) - \frac{\kappa_0 T}{U} \sum_{i=1}^U \frac{\ln(1/\delta_i)}{\epsilon_i^2} - \frac{\kappa_0 T}{U \ln A} \sum_{i=1}^U \frac{\ln(1/\delta_i)}{\epsilon_i^2} \right). \quad (25)$$

If U is set to be large enough, and $\sum_{i=1}^U \frac{\ln(1/\delta_i)}{\epsilon_i^2}$ is small, we have $\frac{\partial^2 h(U, T, \epsilon_i)}{\partial T^2} > 0$ and the upper bound is convex for T . \square

In more detail, a larger T has a negative impact on the model quality by increasing the amount of noise added in each communication round for a given $\epsilon_i, \forall i \in \mathcal{U}$ (in line with (23)), but it also has a positive impact on the convergence because it reduces the loss function value with more iterations.

According to our analysis above, there exists an optimal value of T for given privacy levels $\epsilon_i, \forall i \in \mathcal{U}$. However, this optimal value cannot be derived directly, since some parameters of the loss function are difficult to obtain accurately. One possible method for obtaining the optimal T is through exhaustive search, i.e., try different value of T and choose the one with the highest convergence performance as the practical value in use. The time complexity of the exhaustive search method is determined by the searching interval and a smaller interval means a larger complexity but a higher performance. Hence, exhaustive search is time-consuming and computationally complex. In the next section, we will propose an efficient algorithm for finding a good value of T to achieve a high convergence performance.

4.2 Proposed Discounting Communication Rounds

Based on the analysis above, we can note that an improper value of T will damage the performance of the UDP algorithm. Hence, if we reduce the value of T slightly when the training performance stops improving, we can obtain a smaller STD as well as improving the training performance. Therefore, we design a CRD algorithm by adjusting the number of communication rounds T with a discounting method during the training process to achieve a better

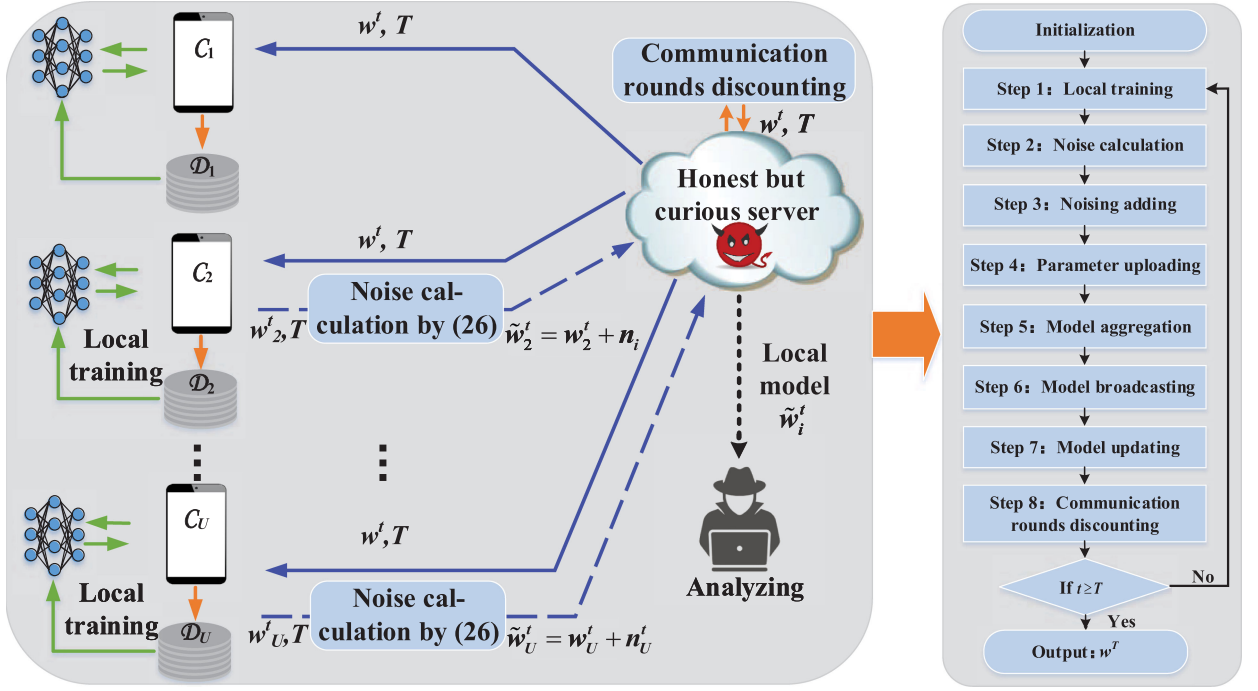


Fig. 1. The training process of our proposed UDP algorithm with CRD method at the t th communication round. As shown in the figure, we have 8 steps in each communication round. Our algorithm will recalculate the value of T (the updated T is generally less than the previous one) when the training performance stops improving. The advantages of our algorithm are as follows. First, we can obtain a smaller T , i.e., less training time, compared with the conventional UDP-based FL. Second, we can achieve a better convergence performance (due to less noise added to the model) while keeping the DP level unchanged.

convergence performance. The training process of such a CRD algorithm in UDP contains following steps:

- **Initialization:** The server broadcasts the initial parameters (i.e., w^0 and T) to all MTs;
- **Step 1: Local training:** All active MTs locally compute training parameters with local datasets and the global parameter. In order to prove the DP guarantee, the influence of each individual example on local gradients should be bounded with a clipping threshold C . Each gradient vector will be clipped in L_2 norm, i.e., the i th local gradient vector g_i^t at the t th communication round is replaced by $g_i^t / \max(1, \|g_i^t\|)$. We can remark that parameter clipping of this form is a popular ingredient of ML for non-privacy reasons;
- **Step 2: Noise calculation:** Each MT obtain the STD σ_i of artificial Gaussian noise using the proposed noise calculation method (introduced in the following subsection);
- **Step 3: Noise adding:** Each MT add artificial Gaussian noise with a certain STD to the local trained parameters in order to guarantee (ϵ_i, δ_i) -LDP;
- **Step 4: Parameter uploading:** All active MTs upload the noised parameters to the server for aggregation;
- **Step 5: Model aggregation:** The server performs aggregation over the uploaded parameters from MTs;
- **Step 6: Model broadcasting:** The server broadcasts the aggregated parameters and the number of communication rounds T to all MTs;
- **Step 7: Model updating:** All MTs update their respective models with the aggregated parameters, then

test the performance of the updated models and upload the performance to the server;

- **Step 8: Communication rounds discounting:** When the convergence performance stops improving by the following decision $\mathcal{V}(w^t) - \mathcal{V}(w^{t+1}) < \zeta$ and ζ is the threshold, the discounting method will be triggered in the server, where $\mathcal{V}(w^t)$ is the test loss by the model w^t . The server will obtain a smaller T than the previous one with a linear discounting factor β and an integer value by $T = \lfloor \beta(T - t) \rfloor + t$. This factor can control the decaying speed of T . The FL process will be completed when the aggregation time reaches the preset T .

First, we can note that the value of ζ will have an impact on the number of communications T and not affect the convergence and the privacy guarantee directly. Then, the value of ζ is to predict whether the training process is stopping and determine when to trigger the discounting method (adjust the number of communications). In the conventional machine learning, a threshold is common to predict whether the training process is stopping. Therefore, in our experiments, we adopt a small positive value 0.001 as the value of ζ . Moreover, in this method, the value of T is determined iteratively to ensure a high convergence performance in FL training. Obviously, when the value of T is adjusted, we must calculate a new STD of additive noises in terms of previous training process. The diagrammatic expression of this method is shown in Fig. 1. Therefore, we will develop a noise calculation method to update the STD of additive noises and T alternately in the following subsection.

4.3 Noise Calculation for Varying T

Now, let t be the index of the current communication round and σ_i^τ ($0 \leq \tau \leq t-1$) be the STD of additive noises for the i th MT at the τ th communication round. In our CRD algorithm, with a new T , if t is greater than T , the training process will stop. If t is less than T , we need to calculate the STD of noises and add them on the local parameters in the following communication round. Considering this, we obtain the following theorem.

Theorem 4. After t ($0 \leq t < T$) communication rounds and a new T , the STD of additive noises for the i th MT to guarantee (ϵ_i, δ_i) -LDP can be given as

$$\sigma_i^t = \left(\frac{T-t}{\frac{\epsilon_i^2}{2q\Delta\ell^2 \ln\left(\frac{1}{\delta_i}\right)} - \sum_{\tau=0}^{t-1} \frac{1}{(\sigma_i^\tau)^2}} \right)^{\frac{1}{2}}. \quad (26)$$

Proof. Using the above definition of moments (17) and a preset value of T , we have

$$\begin{aligned} \alpha^T(\lambda) &\leq \sum_{\tau=1}^T \alpha(\lambda) = \sum_{\tau=1}^t \frac{q\lambda(\lambda+1)\Delta\ell^2}{2(\sigma_i^\tau)^2} \\ &\quad + \sum_{\tau=1}^{T-t} \frac{q\lambda(\lambda+1)\Delta\ell^2}{2\sigma_i^2} \\ &= \sum_{\tau=1}^t \frac{q\lambda(\lambda+1)\Delta\ell^2}{2(\sigma_i^\tau)^2} + \frac{(T-t)q\lambda(\lambda+1)\Delta\ell^2}{2\sigma_i^2}. \end{aligned} \quad (27)$$

Using the tail bound by moments [3], we have

$$\delta = \min_{\lambda} \exp(\alpha^T(\lambda) - \lambda\epsilon_i). \quad (28)$$

Because of

$$\begin{aligned} \min\{\alpha^T(\lambda) - \lambda\epsilon_i\} &= -\frac{q\Delta\ell^2}{8} \left(\sum_{\tau=1}^t \frac{1}{(\sigma_i^\tau)^2} + \frac{T-t}{\sigma_i^2} \right) \\ &\quad + \frac{\epsilon_i}{2} - \frac{\epsilon_i^2}{2q\Delta\ell^2 \left(\sum_{\tau=1}^t \frac{1}{(\sigma_i^\tau)^2} + \frac{T-t}{\sigma_i^2} \right)}, \end{aligned} \quad (29)$$

where $\lambda = -\frac{1}{2} + \frac{\epsilon_i\sigma_i^2}{q\Delta\ell^2 \left(\sum_{\tau=1}^t \frac{1}{(\sigma_i^\tau)^2} + \frac{T-t}{\sigma_i^2} \right)}$. Therefore, we have

$$\begin{aligned} \ln\left(\frac{1}{\delta_i}\right) &\leq \frac{q\Delta\ell^2}{8} \left(\sum_{\tau=1}^t \frac{1}{(\sigma_i^\tau)^2} + \frac{T-t}{\sigma_i^2} \right) - \frac{\epsilon_i}{2} \\ &\quad + \frac{\epsilon_i^2}{2q\Delta\ell^2 \left(\sum_{\tau=1}^t \frac{1}{(\sigma_i^\tau)^2} + \frac{T-t}{\sigma_i^2} \right)} \\ &< \frac{\epsilon_i^2}{2q\Delta\ell^2 \left(\sum_{\tau=1}^t \frac{1}{(\sigma_i^\tau)^2} + \frac{T-t}{\sigma_i^2} \right)}. \end{aligned} \quad (30)$$

Then, we can set (26) to guarantee (ϵ_i, δ_i) -DP for the following training. \square

In Theorem 4, we can obtain a proper STD of the additive noises based on the previous training process and the value of T . From this result, we can find that if we have large STDs (strong privacy guarantee) in the previous $t-1$ training processes, i.e., σ_i^τ is large, the calculated STD will be small (weak privacy guarantee), i.e., σ_i^t is small.

We can also note that if the value of T is not changed in this communication round, the value of STD will remain unchanged. Considering σ_i^t , σ_i^{t+1} and unchanged T , from Equation (26), we can obtain

$$\sigma_i^{t+1} = \left(\frac{T-t-1}{\frac{\epsilon_i^2}{2q\Delta\ell^2 \ln\left(\frac{1}{\delta_i}\right)} - \sum_{\tau=0}^t \frac{1}{(\sigma_i^\tau)^2}} \right)^{\frac{1}{2}}, \quad (31)$$

and

$$\frac{\epsilon_i^2}{2q\Delta\ell^2 \ln\left(\frac{1}{\delta_i}\right)} - \sum_{\tau=0}^{t-1} \frac{1}{(\sigma_i^\tau)^2} = \frac{T-t}{(\sigma_i^t)^2}. \quad (32)$$

Substituting Equation (32) into Equation (31), we have $\sigma_i^{t+1} = \sigma_i^t$, which is in line with our analysis. In this case, we summarize the detailed steps of the UDP with CRD method in Algorithm 2.

Algorithm 2. UDP With CRD Method

Input: The value of an initial T , LDP parameters (ϵ_i, δ_i) , clipping threshold C and discounting factor β ($\beta < 1$).

- 1: Initialize: $t = 0$ and w^0
 - 2: **while** $t < T$ **do**
 - 3: Broadcast: w^t and T to all MTs
 - 4: **for** $\forall i \in \mathcal{K}$ **do**
 - 5: Calculate the STD of additive noises using (26);
 - 6: Locally train with clipping gradients:
 - 7: $w_i^{t+1} = \ell(\mathcal{D}_i, w^t)$;
 - 8: Add (ϵ_i, δ_i) -LDP noise:
 - 9: $\tilde{w}_i^{t+1} = w_i^t + \mathcal{N}(0, \sigma_i^t \mathbf{I})$;
 - 10: Upload noised parameters to the server;
 - 11: Aggregate received model parameters:
 - 12: $w^{t+1} = \sum_{i \in \mathcal{K}} p_i \tilde{w}_i^{t+1}$;
 - 13: **for** $\mathcal{C}_i \in \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_U\}$ **do**
 - 14: Test the aggregating parameters w^{t+1} ;
 - 15: using local dataset
 - 16: **if** $\mathcal{V}(w^t) - \mathcal{V}(w^{t+1}) < \zeta$ **then**
 - 17: Update the preset value of T :
 - 18: $T = \lfloor \beta(T-t) \rfloor + t$;
 - 19: $t = t + 1$;
 - 20: **return** w^T
-

4.4 Complexity Analysis

The main difference between the proposed training protocol in Algorithm 1 and the conventional algorithm is the gradients clipping, noise computing and noise adding. During the training process in Algorithm 1, each MT needs to clip the gradients by the equation $g_{i,m}^t(\mathcal{D}_{i,m}) = g_{i,m}^t(\mathcal{D}_{i,m}) / \max(1, \frac{\|g_{i,m}^t(\mathcal{D}_{i,m})\|_2}{C})$ locally, which takes $O(s(w))$ time, where $s(w)$ is the size of w . We can note that this clipping process will compare the norm of gradients and the

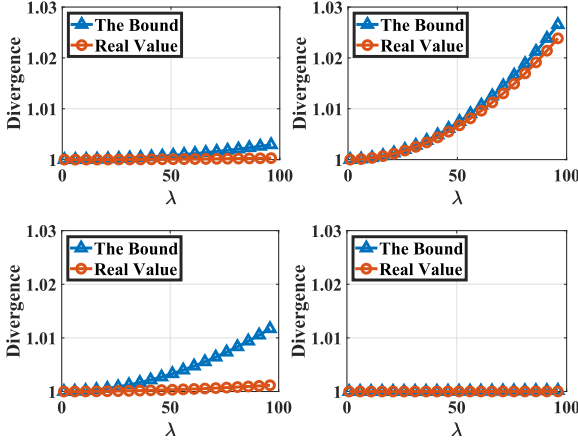


Fig. 2. The comparison of our bound and real value of divergence in (16). (a) $q = 0.9$, $|\mathcal{D}_i| = 800$, $U = 50$. (b) $q = 0.1$, $|\mathcal{D}_i| = 800$, $U = 50$. (c) $q = 0.9$, $|\mathcal{D}_i| = 400$, $U = 50$. (d) $q = 0.9$, $|\mathcal{D}_i| = 800$, $U = 200$.

clipping threshold C to bound the gradients. The noise computing can be expressed as $\sigma_i = \frac{\Delta \ell \sqrt{2qT \ln(1/\delta_i)}}{\epsilon_i}$ and takes $O(1)$ time, which can be obtained directly when all required parameters are ready. Finally, the noise adding is given by $\tilde{w}_i^{t+1} = w_i^{t+1} + \mathcal{N}(0, \sigma_i^2 \mathbf{I})$, where the complexity is governed by the size of model parameters and is $O(s(w))$.

Meanwhile, Algorithm 2 has improved the noise computing and added the adjusting process for the number of communication rounds (T) compared with Algorithm 1. Therefore, the noise computing in Algorithm 2 can be given by $\sigma_i^t = (T - t)^{\frac{1}{2}} / \left(\frac{\epsilon_i^2}{2q\Delta \ell^2 \ln(\frac{1}{\delta_i})} - \sum_{\tau=0}^{t-1} \frac{1}{(\sigma_i^\tau)^2} \right)^{\frac{1}{2}}$ and obtained directly when all required parameters are ready. Moreover, the adjusting process for the number of communication rounds T only needs few computing resources. Both processes only need $O(1)$ time.

In summary, the time complexity analysis of the proposed training protocols, i.e., Algorithms 1 and 2, is $O(s(w))$. Furthermore, the proposed training protocols consume few more computing resources.

5 EXPERIMENTAL RESULTS

In this section, we evaluate the accuracy of our analytical results for different learning tasks. Then, we evaluate our proposed CRD method in UDP algorithm, and demonstrate the effectiveness of various parameter settings, such as the privacy level, the initial value of T and discounting factor.

5.1 Evaluation of Numerical Results

In this subsection, we first describe our numerical validation of the bound compared with the real value in (16) with $\sigma_i = 0.01$ by varying λ from 1 to 100. In Fig. 2a, we set $q = 0.9$, $|\mathcal{D}_i| = 800$, $\forall i$, and $U = 50$. In our validation, our bound is close but always higher than the real divergence. We also did tests for cases such as a smaller sampling ratio $q = 0.1$, a smaller size of local datasets $|\mathcal{D}_i| = 400$, $\forall i$, and a larger number of MTs $U = 200$ in Figs. 2b, 2c and 2d, respectively. We find that the empirical bound always holds and closes under the given conditions, especially for a large enough $|\mathcal{D}_i|$. Therefore, we have the conjecture that our bound is a valid for the analytical moments accountant of sampled Gaussian mechanism and seek its formal proof in our future work.

5.2 Experimentation Setup

We evaluate the training of three different machine learning models on different datasets, namely support vector machine (SVM) on the IPUMS-US dataset, multi-layer perceptron (MLP) on the standard MNIST dataset and the convolutional neural network (CNN) on the CIFAR-10 dataset, respectively.

Models and Datasets Description. The models include SVM, MLP and CNN, which are detailed as follows.

1) SVM is trained on the IPUMS-US dataset, which are census data extracted from [18] and contain 40000 individual records with 58 attributes including age, education level and so on. The categorical attributes in the dataset are denoted by various integers. The label of each sample describes whether the annual income of this individual is over 25k. In this model, the loss function is given by

$$F(w) = \frac{\kappa}{2} \|w\|_2^2 + \max\{y_m - w^\top \mathcal{D}_{i,m}, 0\}, \quad (33)$$

where $\kappa > 0$ is a regularization coefficient, $\mathcal{D}_{i,m}$ is the m th sample in \mathcal{D}_i , \mathcal{D}_i is the dataset of i th MT and $y_m \in \{+1, -1\}$ for $m \in \{1, \dots, |\mathcal{D}_i|\}$.

2) MLP is trained using SGD consisting of single hidden layer with 256 hidden units, where ReLU units and softmax of 10 classes (corresponding to the 10 digits) are applied. We use the cross-entropy loss function and conduct experiments on the standard MNIST dataset for handwritten digit recognition consisting of 60000 training examples and 10000 testing examples [28]. Each example is a 28×28 size gray-level image of handwritten digits from 0 to 10. In Fig. 3, we show several samples of the standard MNIST dataset with a visual illustration via MLP. The left figure and right figure in Fig. 3 are derived from the original FL model and UDP

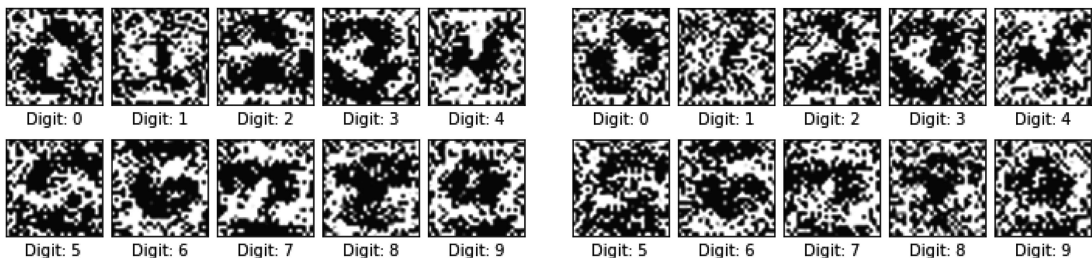


Fig. 3. Visual illustration of the standard MNIST dataset via MLP, in which the left is original FL model and the right is the model trained by UDP algorithm. Basically speaking, these subfigures show the typical digits that are learned by the machine models with the original FL generates less noisy digits than the UDP based FL.

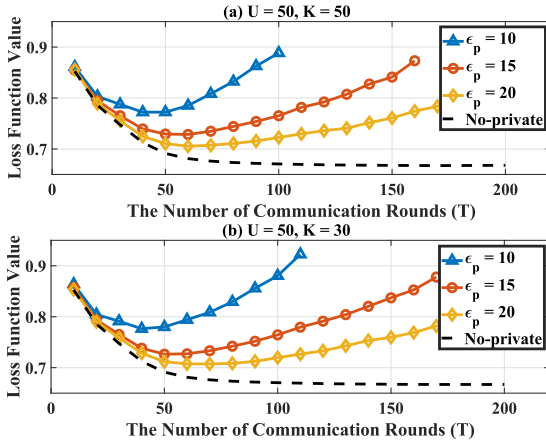


Fig. 4. Value of the loss function under various T using the UDP algorithm with the SVM model. (a) $U = 50, K = 50$ ($q = 1$). (b) $U = 50, K = 30$ ($q = 0.6$).

model, respectively. The visual results of MLP based FL with the interpretability technique are corresponding to the digit 0 \sim 9 under the original FL and UDP based FL, respectively. Basically speaking, these subfigures show that a digit understood by the machine model with the original FL are more distinct than the UDP based FL.

3) CNN is trained using SGD consisting of single convolutional layer with the convolutional kernel size 5 and the padding size 4, where ReLU units and softmax of 10 classes are applied. We also use the cross-entropy loss function in this model. The CIFA-10 dataset consists of 32×32 color images with three channels (RGB) in 10 classes including ships, planes, dogs and cats. Each class has 6000 images where there are 40000 examples for training, 10000 for testing and 10000 for validation.

Among them, the loss function for SVM is convex, whereas the loss function for MLP and CNN does not satisfy this condition. The experimental results in this setting show that our theoretical results and proposed algorithm also work well for models (such as MLP and CNN) whose loss functions are not convex.

Parameter Setting. The total number of MTs U in our experiments is set to 50. In order to conduct our experiments conveniently, we consider the worst condition for model convergence with $\epsilon_i = \epsilon_p$ and $\delta_i = \delta_p, \forall i \in \mathcal{U}$,

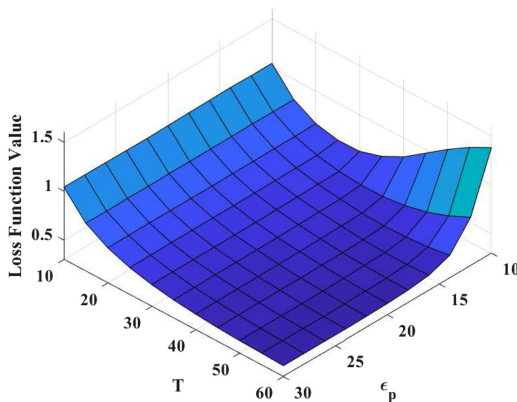


Fig. 5. Value of the loss function under various T and privacy levels using the MLP model based UDP algorithm with $U = K = 50$ ($q = 1$).

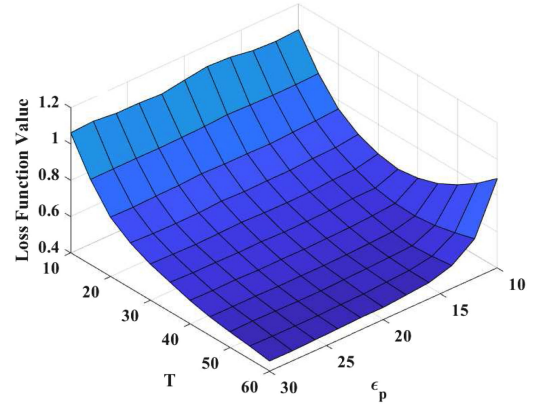


Fig. 6. Value of the loss function under various T and privacy levels using the MLP model based UDP algorithm with $U = 50$ and $K = 30$ ($q = 0.6$).

where ϵ_p and δ_p are the smallest values of permitting. In addition, we adopt a certain δ to study the effect of ϵ_p and set $\delta_p = 0.001$. The threshold ζ in CRD method is set to 0.001. We can note that parameter clipping C is a popular ingredient of SGD and ML for non-privacy reasons. A proper value of clipping threshold C should be considered for the DP based FL framework. In the following experiments, we utilize the method in [29] and choose C by taking the median of the norms of the unclipped parameters over the course of training.

5.3 Effects of the Number of Communication Rounds on UDP

In this subsection, we verify the convex property of UDP for the value of T with SVM and MLP models. In Fig. 4, we show experimental results of testing loss as a function of T with various privacy levels using SVM. The size of local samples is set as $|\mathcal{D}_i| = 128$. The observation is in line with Theorems 2 and 3, and the reason comes from the fact that a lower privacy level decreases the standard deviation of additive noises and the server can obtain better quality ML model parameters from MTs. Fig. 4 also implies that an optimal value of T increases with a larger ϵ_p .

Then, we show experimental results of the loss function value with respect to T and privacy levels ϵ_p using the MLP network. The size of local samples is set as $|\mathcal{D}_i| = 800$ in this experiment. Figs. 5 and 6 illustrate the expectation of the loss function, by varying privacy level ϵ_p and the value of T . From Fig. 5, we can observe that a large T and a small ϵ_p may lead to a worse performance in the no sampling scenario. As shown in the second scenario with sampling ratio $q = 0.6$ in Fig. 6, it also retains the same property.

5.4 Effects of Parameters on CRD

In this subsection, to evaluate our CRD algorithm, we apply the MLP model with the standard MNIST dataset. Several experimental settings in our case study are introduced as follows. The setting including the following main parts: 1) various initial values of T using discounting method; 2) various privacy levels using CRD algorithm; and 3) various discounting factors using CRD algorithm.

We evaluate the effectiveness of our CRD method, and compare the results with the following benchmarks: 1)

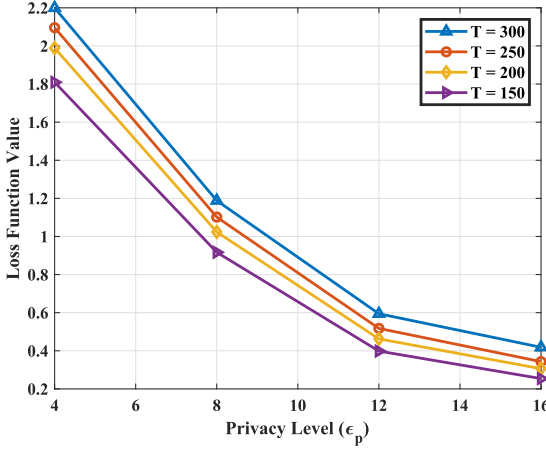


Fig. 7. Value of loss function by a MLP based FL using the UDP algorithm with CRD method ($\beta = 0.9$) under various initial values of T .

original UDP, in which uniform privacy budget (ϵ_p) allocation algorithm as well as the moments accountant method [3] is adopted in the UDP algorithm; 2) *Lei's approach* [29] in which the STD of added noise will be reduced linearly until the privacy loss is larger than a preset ϵ_p ; and 3) *optimal UDP*, in which the UDP algorithm will be trained with the optimal T (obtained by heuristic search).

Initial Value of T . In the previous experiments, the initial value of T is set as the default. To examine the effect of initial T , we vary its value from 150 to 300 and measure the model convergence performance under several fixed privacy levels. We also choose this handwritten digit recognition task with the size of local samples $|\mathcal{D}_i| = 800$ and the discounting factor $\beta = 0.9$. Fig. 7 shows that when T is closer to the optimal value of T (the optimal value of T by searching is shown in the above subsection), we can obtain a better convergence performance. This observation is also in line with Theorems 2 and 3.

Privacy Level. We choose a handwritten digit recognition task with the initial number of communication rounds $T = 200$ and the size of local samples $|\mathcal{D}_i| = 800$. We also set two different sampling ratio $q = 1$ and $q = 0.6$, which are corresponding to Fig. 8, observed by (a) and (b), respectively. In Fig. 8, we describe how value of the loss function change

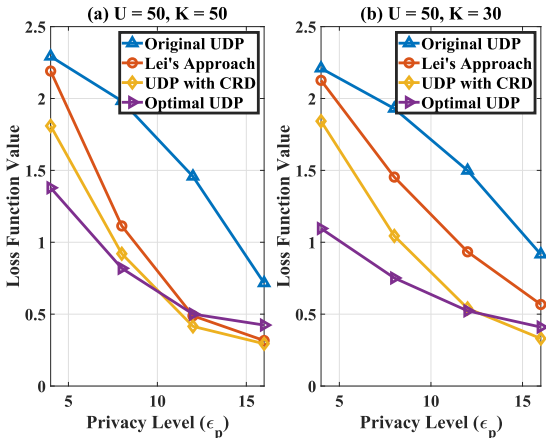


Fig. 8. Value of the loss function by a MLP based FL using the original UDP, Lei's approach, UDP with CRD method ($\beta = 0.9$) and the optimal UDP. (a) $U = 50, K = 50$. (b) $U = 50, K = 30$.

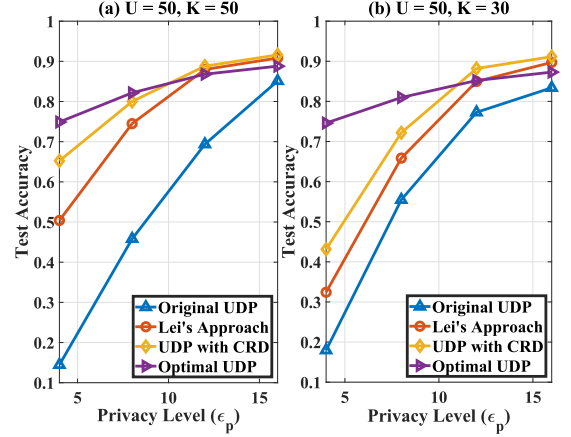


Fig. 9. Test accuracy by a MLP based FL using the original UDP, Lei's approach, UDP with CRD method ($\beta = 0.9$) and the optimal UDP. (a) $U = 50, K = 50$. (b) $U = 50, K = 30$.

with various values of the privacy level ϵ_p under original UDP (Algorithm 1), UDP with CRD (Algorithm 2) and optimal CRD (obtain the optimal T by searching). Fig. 9 shows the test accuracy corresponding to Fig. 8. We can note that using Algorithm 2 with discounting factor $\beta = 0.9$ can greatly improve the convergence performance in Figs. 8a and 8b, which is close to the optimal results. Moreover, our UDP with CRD has a better performance than the optimal results with a large ϵ_p , because the UDP algorithm with the optimal T (obtained by searching) averagely allocates the privacy budget (ϵ_p) during the training process but our algorithm can utilize the privacy budget more adaptively and efficiently. In Fig. 10, we choose the same parameters with Fig. 8 and show the change of T during training in one experiment under the UDP algorithm with CRD method ($\beta = 0.9$). In Fig. 10, we can find that a smaller privacy level ϵ_p will have an early end in UDP algorithm with CRD method. The intuition is that a larger T can lead to a higher chance of information leakage and a larger noise STD of additive noises. Then, the CRD method may be triggered and a decreased T will be broadcasted to chosen MTs from the server.

In Fig. 11, we evaluate a CNN based federated learning for the multi-class classification task with the dataset CIFAR-10 in UDP with CRD, where each client has 800

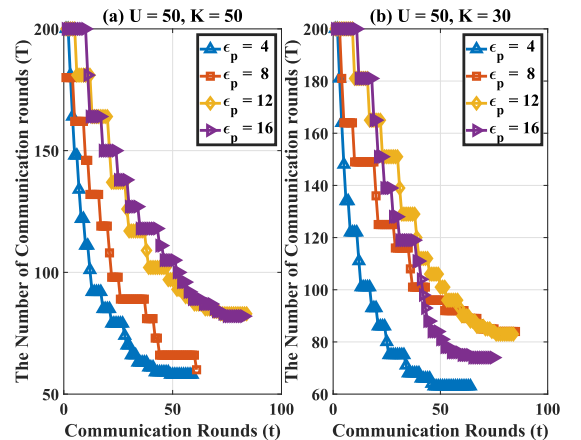


Fig. 10. The number of communication rounds by a MLP based FL using UDP algorithm with CRD method ($\beta = 0.9$). (a) $U = 50, K = 50$. (b) $U = 50, K = 30$.

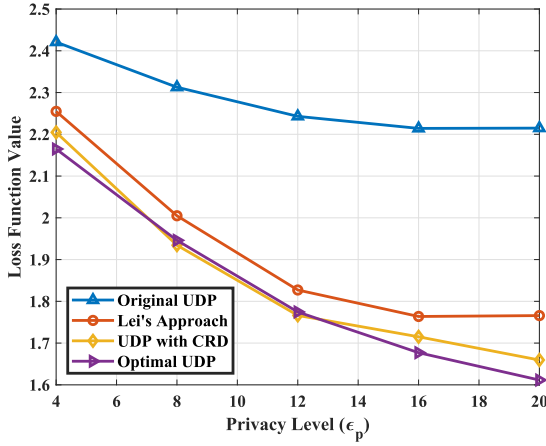


Fig. 11. Value of the loss function using the original UDP, Lei's approach, UDP with CRD method ($\beta = 0.9$) and the optimal UDP by a CNN based FL for the multi-class classification task with the dataset CIFAR-10.

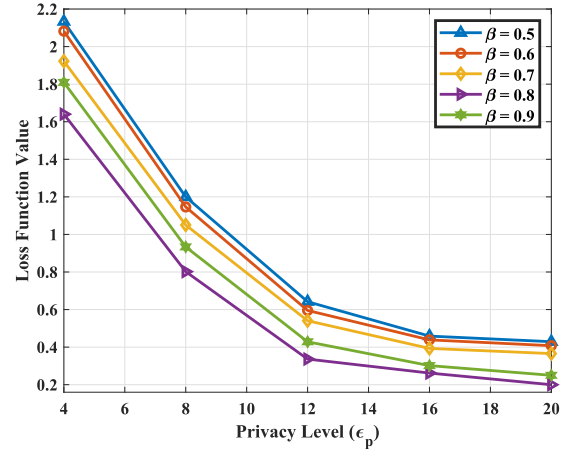


Fig. 14. Value of loss function by a MLP based FL using the UDP algorithm with CRD method under various discount factors.

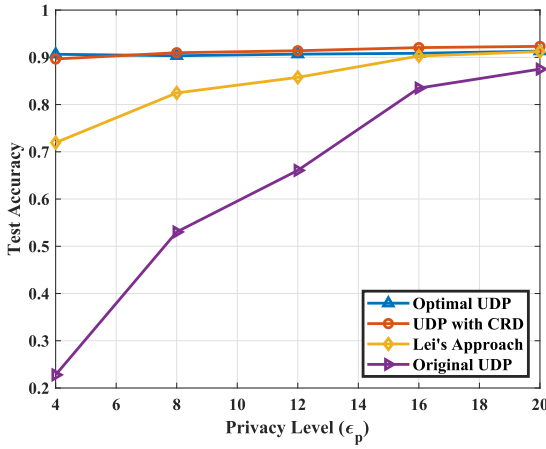


Fig. 12. Test accuracy by a MLP based FL using the original UDP, Lei's approach, UDP with CRD method ($\beta = 0.9$) and the optimal UDP under the non-IID data distribution.

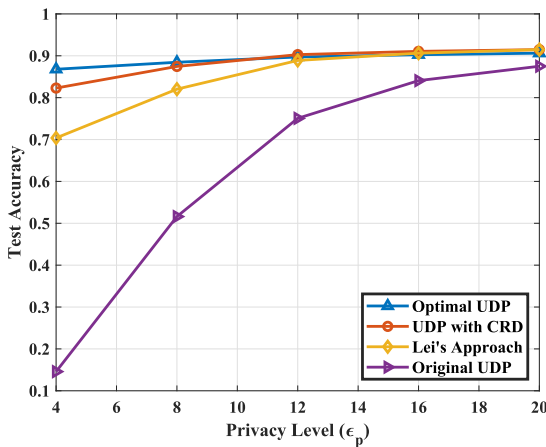


Fig. 13. Test accuracy by a MLP based FL using the original UDP, Lei's approach, UDP with CRD method ($\beta = 0.9$) and the optimal UDP under the unbalanced setting (different number of samples for different MTs).

training samples locally. The protection levels are set to $\epsilon = 4$, $\epsilon = 8$, $\epsilon = 12$, $\epsilon = 16$ and $\epsilon = 20$ for this experiment. In addition, we set $N = 50$, $K = 30$, $T = 200$ and $\beta = 0.9$. From

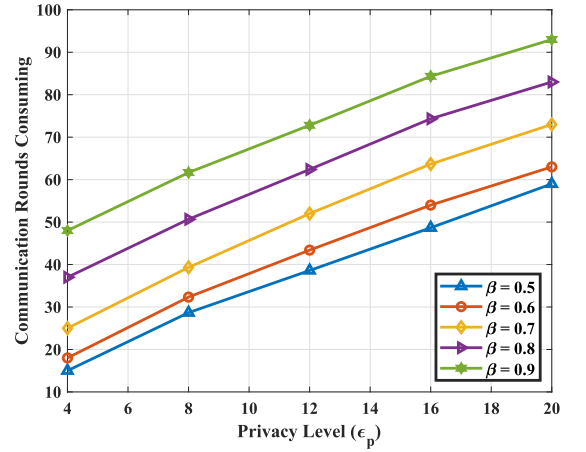


Fig. 15. Communication rounds consuming by a MLP based FL using the UDP algorithm with CRD method under various discount factors.

Fig. 11, comparing with the original UDP and Lei's approach, we can note that using Algorithm 2 with discounting factor $\beta = 0.9$ can greatly improve the convergence performance.

We also apply the MLP model with the standard MNIST dataset with the settings of non-IID data distribution and different number of samples (unbalanced) in UDP with CRD. For the non-IID data distribution setting, each MT has four kinds of digits with the same amount, and the variety is different from all other MTs. In the unbalanced setting, all MTs are divided into 5 parts and the MT in each part has different number of training samples (400, 600, 800, 1,000 and 1,200, respectively) locally. The protection levels are set to $\epsilon = 4$, $\epsilon = 8$, $\epsilon = 12$, $\epsilon = 16$ and $\epsilon = 20$ for this experiment. In addition, we set $N = K = 50$, $T = 200$ and $\beta = 0.9$. From Figs. 12 and 13, we can note that using Algorithm 2 with discounting factor $\beta = 0.9$ can greatly improve the test accuracy comparing with the original UDP and Lei's approach.

Discounting Factor. In Fig. 14, we vary the value of β from 0.5 to 0.9 and plot their convergence results of the loss function. The size of local samples and the initial number of communication rounds are set as $|D_i| = 800$ and $T = 200$, respectively. We observe that when the privacy level is fixed, a larger β results in a slower decay speed of the T

which means careful adjustments of T in the training and will benefit the convergence performance. This is consistent with Theorems 2 and 3. With various values of different β , when we choose $\beta = 0.8$, the UDP algorithm with CRD method will have the best convergence performance. In Fig. 15, we show the communication rounds consuming (the number of required communication rounds) with various discounting factors using the same parameters with Fig. 14. We find that more careful adjustments (corresponding to a larger β) will lead to more communication rounds consuming. Hence, we can conclude that there is a tradeoff between communication rounds consuming and convergence performance by choosing β . As a future work, it is of great interest to analytically evaluate the optimum value of β to minimize the loss function.

6 CONCLUSION

In this paper, we have introduced a UDP algorithm in FL to preserve MTs' privacy and proved that the UDP algorithm can satisfy the requirement of LDP under a certain privacy level by properly selecting the STD of additive noise processes. Then, we have shown that there is an optimal number of communication rounds (T) in terms of convergence performance for a given protection level, which has motivated us to design an intelligent scheme for adaptively choosing the value of T . To address this problem, we have proposed a CRD method for training an FL model, which will be triggered when the convergence performance stops improving. Our experiments have demonstrated that this discounting method in the UDP algorithm can obtain a better tradeoff between convergence performance and privacy levels. We have also noted that various initial values of T and discounting factors β bring out different convergence results.

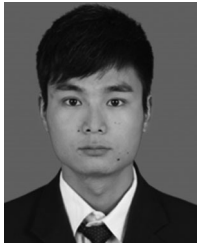
It is noteworthy that the privacy budget allocation scheme greatly affects the quality of the FL training and the proposed CRD method also can be improved if we can obtain an exact FL performance prediction. As a topic for future, it is of interest to design an effective privacy budget allocation scheme to improve the convergence performance with a given privacy level.

ACKNOWLEDGMENTS

This work was supported in part by National Key Research and Development Program under Grant 2018YFB1004800, in part by National Natural Science Foundation of China under Grants 61872184 and 62002170, and in part by the U.S. National Science Foundation under Grant CCF-1908308.

REFERENCES

- [1] J. Li, Y. Chen, Z. Lin, W. Chen, B. Vucetic, and L. Hanzo, "Distributed caching for data dissemination in the downlink of heterogeneous networks," *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3553–3568, Oct. 2015.
- [2] X. Deng, J. Li, L. Shi, Z. Wei, X. Zhou, and J. Yuan, "Wireless powered mobile edge computing: Dynamic resource allocation and throughput maximization," *IEEE Trans. Mobile Comput.*, early access, Oct. 28, 2020, doi: [10.1109/TMC.2020.3034479](https://doi.org/10.1109/TMC.2020.3034479).
- [3] A. Martin *et al.*, "Deep learning with differential privacy," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.
- [4] L. Fang, W. Susilo, C. Ge, and J. Wang, "Public key encryption with keyword search secure against keyword guessing attacks without random oracle," *Inf. Sci.*, vol. 238, pp. 221–241, 2013.
- [5] S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, and J. Li, "Privacy preserving location data publishing: A machine learning approach," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 7, 2020, doi: [10.1109/TKDE.2020.2964658](https://doi.org/10.1109/TKDE.2020.2964658).
- [6] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, "Secure keyword search and data sharing mechanism for cloud computing," *IEEE Trans. Dependable Secure Comput.*, early access, Jan. 3, 2020, doi: [10.1109/TDSC.2020.2963978](https://doi.org/10.1109/TDSC.2020.2963978).
- [7] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," 2016. [Online]. Available: <http://arxiv.org/abs/1602.05629>
- [8] J. Konečný *et al.*, "Federated learning: Strategies for improving communication efficiency," 2016. [Online]. Available: <http://arxiv.org/abs/1610.05492>
- [9] C. Ma *et al.*, "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, Jul./Aug. 2020.
- [10] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Trans. Mobile Comput.*, early access, May 14, 2020.
- [11] D. C. Nguyen *et al.*, "Enabling AI in future wireless networks: A data life cycle perspective," *IEEE Commun. Surveys Tuts.*, early access, Sep. 18, 2020, doi: [10.1109/COMST.2020.3024783](https://doi.org/10.1109/COMST.2020.3024783).
- [12] Z. Wang *et al.*, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2019, pp. 2512–2520.
- [13] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Security Privacy*, 2019, pp. 691–706.
- [14] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning: Revisited and enhanced," in *Proc. Int. Conf. Appl. Techn. Inf. Secur.*, 2017, pp. 100–110.
- [15] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [16] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 14 774–14 784.
- [17] C. Ma *et al.*, "RDP-GAN: A Rényi-differential privacy based generative adversarial network," 2020. [Online]. Available: <http://arxiv.org/abs/2007.02056>
- [18] J. Lee and D. Kifer, "Concentrated differentially private gradient descent with adaptive per-iteration privacy budget," in *Proc. ACM Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1656–1665.
- [19] C. Li, P. Zhou, L. Xiong, Q. Wang, and T. Wang, "Differentially private distributed online learning," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1440–1453, Aug. 2018.
- [20] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "DP-ADMM: ADMM-based distributed learning with differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1002–1012, 2020.
- [21] N. Wu, F. Farokhi, D. Smith, and M. A. Kaafer, "The value of collaboration in convex machine learning with differential privacy," in *Proc. IEEE Symp. Security Privacy*, 2020, pp. 304–317.
- [22] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.
- [23] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2017. [Online]. Available: <http://arxiv.org/abs/1712.07557>
- [24] S. Truex *et al.*, "A hybrid approach to privacy-preserving federated learning," in *Proc. ACM Workshop Artif. Intell. Secur.*, 2019, pp. 1–11.
- [25] T. Li, Z. Liu, V. Sekar, and V. Smith, "Privacy for free: Communication-efficient learning with differential privacy using sketches," 2019. [Online]. Available: <https://arxiv.org/abs/1911.00972>
- [26] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Security Privacy*, 2017, pp. 3–18.
- [27] S. Wang *et al.*, "Local differential private data aggregation for discrete distribution estimation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 2046–2059, Sep. 2019.
- [28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [29] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *Proc. IEEE Symp. Security Privacy*, 2019, pp. 332–349.



Kang Wei (Student Member, IEEE) received the BSc degree in information engineering from Xidian University, Xian, China, in 2014, and the MSc degree from the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2018, where he is currently working toward the PhD degree. His current research interests include data privacy and security, differential privacy, AI and machine learning, information theory, and channel coding theory in NAND flash memory.



Jun Li (Senior Member, IEEE) received the PhD degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, P. R. China, in 2009. From January 2009 to June 2009, he worked with the Department of Research and Innovation, Alcatel Lucent Shanghai Bell as a research scientist. From June 2009 to April 2012, he was a postdoctoral fellow with the School of Electrical Engineering and Telecommunications, the University of New South Wales, Australia. From April 2012 to June 2015, he was a research

fellow with the School of Electrical Engineering, the University of Sydney, Australia. From June 2015, he is a professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. He was a visiting professor with Princeton University, Princeton, New Jersey from 2018 to 2019. His research interests include network information theory, game theory, distributed intelligence, multiple agent reinforcement learning, and their applications in ultra-dense wireless networks, mobile edge computing, network privacy and security, and industrial Internet of Things. He has co-authored more than 200 papers in IEEE journals and conferences, and holds one US patents and more than 10 Chinese patents in these areas. He was serving as an editor of the *IEEE Communication Letters* and TPC member for several flagship IEEE conferences. He received Exemplary Reviewer of *IEEE Transactions on Communications*, in 2018, and Best Paper Award from IEEE International Conference on 5G for Future Wireless Networks, in 2017.



Ming Ding (Senior Member, IEEE) received the BS and MS degrees (with first class Hons.) in electronics engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, and the doctor of philosophy (PhD) degree in signal and information processing from Shanghai Jiao Tong University, China, in 2004, 2007, and 2011, respectively. From April 2007 to September 2014, he worked at Sharp Laboratories of China in Shanghai, China as a researcher/senior researcher/principal researcher. He also served

as the algorithm design director and programming director for a system-level simulator of future telecommunication networks in Sharp Laboratories of China for more than seven years. Currently, he is a senior research scientist at Data61, CSIRO, in Sydney, NSW, Australia. His research interests include information technology, data privacy and security, machine Learning and AI, etc. He has authored more than 100 papers in IEEE journals and conferences, all in recognized venues, and around 20 3GPP standardization contributions, as well as a Springer book "Multi-point Cooperative Communication Systems: Theory and Applications". Also, he holds 21 US patents and co-invented another more than 100 patents on 4G/5G technologies in CN, JP, KR, EU, etc. Currently, he is an editor of the *IEEE Transactions on Wireless Communications* and the *IEEE Wireless Communications Letters*. Besides, he is or has been guest editor/co-chair/co-tutor/TPC member of several IEEE top-tier journals/conferences, e.g., the *IEEE Journal on Selected Areas in Communications*, *IEEE Communications Magazine*, and IEEE Globecom Workshops, etc. He was the lead speaker of the industrial presentation on unmanned aerial vehicles in IEEE Globecom 2017, which was awarded as the Most Attended Industry Program in the conference. Also, he was awarded, in 2017 as the exemplary reviewer for the *IEEE Transactions on Wireless Communications*.



Chuan Ma received the BS degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2013, and the PhD degree from the University of Sydney, Australia, in 2018. He is currently working as a lecturer with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. He has published more than 10 journal and conference papers, including a best paper in WCNC 2018. His research interests include stochastic geometry, wireless caching networks and machine learning, and currently focuses on the big data analysis and privacy preservation.



Hang Su (Member, IEEE) is an academican with the Chinese Academy of Sciences, Beijing, China. He is a foreign member of the Russian academy of natural sciences and is currently dean of the Tsinghua University, Institute for Artificial Intelligence (THUAI), Beijing, China. He is one of the AI pioneers in China, and makes foundational contributions to numerous areas in artificial intelligence including quotient space theory, statistics machine learning, brain-inspired AI, etc.



Bo Zhang (Senior Member, IEEE) is an academican with the Chinese Academy of Sciences, Beijing, China. He is a foreign member of the Russian academy of natural sciences and is currently the dean of the Tsinghua University, Institute for Artificial Intelligence (THUAI), Beijing, China. He is one of the AI pioneers in China, and makes foundational contributions to numerous areas in artificial intelligence including quotient space theory, statistics machine learning, brain-inspired AI, etc. He received the Lifetime Achievement Award of China Computer Federation, in 2014.



H. Vincent Poor (Life Fellow, IEEE) received the PhD degree in EECS from Princeton University, Princeton, New Jersey, in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana-Champaign, Champaign, Illinois. Since 1990 he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University professor of Electrical Engineering. During 2006 to 2016, he served as the dean of Princeton's School of Engineering and Applied Science. He has also held visiting

appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the forthcoming book *Machine Learning and Wireless Communications*. (Cambridge University Press, 2021). He is a member of the National Academy of Engineering and the National Academy of Sciences, and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal and a DEng honoris causa from the University of Waterloo awarded, in 2019.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.