Faithful Edge Federated Learning: Scalability and Privacy

Meng Zhang, *Member, IEEE*, Ermin Wei, *Member, IEEE*, Randall Berry, *Fellow, IEEE*

***Abstract*—Federated learning enables machine learning algorithms to be trained over decentralized edge devices without requiring the exchange of local datasets. Successfully deploying federated learning requires ensuring that agents (e.g., mobile devices) faithfully execute the intended algorithm, which has been largely overlooked in the literature. In this study, we first use risk bounds to analyze how the key feature of federated learning, unbalanced and non-i.i.d. data, affects agents' incentives to voluntarily participate and obediently follow traditional federated learning algorithms. To be more specific, our analysis reveals that agents with less typical data distributions and relatively more samples are more likely to opt out of or tamper with federated learning algorithms. To this end, we formulate the first faithful implementation problem of federated learning and design two faithful federated learning mechanisms which satisfy economic properties, scalability, and privacy. First, we design a *Faithful Federated Learning (FFL) mechanism* which approximates the Vickrey–Clarke–Groves (VCG) payments via an incremental computation. We show that it achieves (probably approximate) optimality, faithful implementation, voluntary participation, and some other economic properties (such as budget balance). Further, the time complexity in the number of agents K is $\mathcal{O}(\log(K))$. Second, by partitioning agents into several clusters, we present a scalable VCG mechanism approximation. We further design a scalable and *Differentially Private FFL (DP-FFL) mechanism*, the first differentially private faithful mechanism, that maintains the economic properties. Our DP-FFL mechanism enables one to make three-way performance tradeoffs among privacy, the iterations needed, and payment accuracy loss.**

***Index Terms*—Federated learning, mechanism design, game theory, differential privacy, faithful implementation.**

I. Introduction

A. Motivation

Machine learning applications often rely on cloud-based datacenters to collect and process the vast amount of needed training data. Due to the proliferation of Internet-of-Things (IoT) applications, much of this data is generated by devices in wireless edge networks. In addition, relatively slow growth in network bandwidth, high latency, and data privacy concerns may make it infeasible or undesirable to upload all the data to a remote cloud, leading some to project that 90% of the global data will be stored and processed locally [2]. *Federated learning* is a nascent solution to retain data in wireless edge networks and perform machine learning training distributively across end-user devices and edge servers (also called edge clouds) (e.g., [3]–[25]).

Federated learning algorithms aim to fit models to data generated by multiple distributed devices. The large-scale deployment of federated learning relies on overcoming the following two main technical challenges: the *statistical challenge* and the *communication challenge* [3]–[5]. Specifically, each agent (the owner of each device) generates data in a non-independently and identically distributed (non-i.i.d.) manner, with the dataset on each device being generated by a distinct distribution and the local dataset size varying greatly. Second, communication is often a significant bottleneck in a federated learning framework, motivating the design of communication-efficient federated learning algorithms. These have been motivating extensive studies on improving efficiency by designing new training models, fast algorithms and quantization techniques (e.g., [12], [13]). Other studies have been designing algorithms for resource allocation, mobile user selection, energy efficient, scheduling, and new communication techniques in wireless edge networks (e.g., [17]–[25]).

Nevertheless, whereas many existing federated learning algorithms assume that agents are obedient, i.e., they are willing to follow the algorithms (e.g., [3]–[9], [12]–[25]), edge devices in practice may be *strategic* and may tamper with or opt out of federated learning algorithms to their own advantages. Such strategic edge devices would be more likely to arise in a wireless setting, where different devices may connect and participate in federated learning at different times. Therefore, the success of deploying federated learning relies on strategic agents' *voluntary participation (into the federation)* and *faithful* execution of distributed federated learning algorithms. Generally speaking, there are two key factors that may incentivize agents to strategically manipulate federated learning:

- federated learning may incur significant resource consumption (e.g., energy, bandwidth, and time) for mobile devices;
- agents have different preferences over prediction outcomes (due to, e.g., non-i.i.d and unbalanced data).

The above issues reflect an agent's dual role as *a contributor* and *a client* in a federated learning setting, respectively. That is, agents demand for both *rewards for contributing* and *preferred prediction outcomes*. This work focuses on the latter issue that has been overlooked in the literature, whereas existing studies mainly have been attempting to solve the former one (see the survey in [26] and references [27]–[40]) by designing incentive mechanisms to reward agents according to agents' data quality, quantity, and reputation. Specifically,

Part of this work has been presented at NetEcon 2021 [1].

This work was supported in part by NSF grants CNS-1908807, AST-2037838, and ECCS-2030251.

Meng Zhang is with the Zhejiang University/University of Illinois at Urbana–Champaign Institute, Zhejiang University, Haining 314400, China (e-mail:jackeymzhang@gmail.com). Part of this work was performed while he was at Northwestern University. Ermin Wei and Randall Berry are with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208 USA (e-mail: ermin.wei@northwestern.edu; rberry@northwestern.edu).

non-i.i.d. and unbalanced data may render different prediction objectives for different individual agents and hence may incentivize strategic manipulation. In this case, each strategic agent can choose either to opt out of or to tamper with the federated learning algorithms to its own advantage. Such a behavior may result in the failure of large-scale deployments of edge federated learning. To this end, this paper will first answer the following question:

Question 1. *How do unbalanced data and non-i.i.d. data distributions disincentivize agents to obediently follow and voluntarily participate into federated learning algorithms?*

To overcome this issue of manipulation, one approach for the server is to leverage *(economic) mechanism design*, by anticipating agents' strategic behaviors. As a seminal example, the Vickrey–Clarke–Groves (VCG) mechanism [41] is a generic truthful mechanism for achieving a socially-optimal solution, while ensuring agents' voluntary participation and incentive compatibility, i.e., truthful reports of their local information. However, many such mechanisms use a central authority that computes the optimal solution, which is not applicable in the framework of federated learning. To achieve distributed implementation of mechanisms, existing studies have developed *faithful* mechanisms (e.g. [42]–[45]), which prevent agents from deviating from the intended algorithms (e.g., by manipulating computation or information reporting).

We note that existing studies on centralized and distributed mechanisms, however, have not addressed the problem of federated learning due to the following important considerations:

- **Scalability**: Mobile devices are expected to be *massively distributed* (i.e., the number of agents may be much larger than the average samples per agent) and have *limited communication* capability [3]–[5]. However, existing VCG-based approaches [41] involve solving $K+1$ optimization problems (where K is the number of agents), which makes them impractical for large-scale systems.
- **Unknown data distributions**: Existing mechanisms assume that agents know their exact objectives, whereas in federated learning agents' expected objectives are unknown to themselves, since their underlying data distributions are unknown.
- **Privacy**: Federated learning often involves training predictive models based on individuals' private local datasets that contain highly sensitive information (e.g., medical records and web browsing history). For instance, multiple hospitals forecast cancer risks by performing federated learning over the whole patient population, while privacy laws prohibit sharing private patient data [9]. However, existing economic mechanisms require strategic agents to reveal their objective values, which may violate such privacy requirements.[1]

These motivate the following key question:

Question 2. *How should one design a faithful federated learning mechanism that also achieves voluntary participation, scalability, and privacy preservation?*

B. Our Work

In light of the challenges above, this paper studies mechanism design for two representative edge federated learning scenarios aiming at achieving scalable, privacy-preserving, and faithful edge federated learning. Similar ideas could be applied to other federated learning algorithms.[2] We summarize our key contributions in the following:

- **Analysis of risk bounds.** We analyze how the key feature in federated learning, non-i.i.d. and unbalanced data, affects agents' incentives to voluntarily participate and obediently follow the algorithms. Specifically, our analysis reveals that an agent with a less typical data distribution and relatively more data samples tends to have a greater incentive to opt out of or tamper with federated learning algorithms.
- **Faithful federated learning.** *We design the first faithful mechanism for federated learning*. It approximates the VCG mechanism and achieves (probably approximate) optimality, faithful implementation, voluntary participation, and some other economic properties (such as budget balance). Further, the time complexity in the number of agents K is $\mathcal{O}(\log(K))$.
- **Differentially private faithful federated learning.** By partitioning agents into several clusters, we present a scalable VCG mechanism approximation with square root iteration complexity. Based on it, we further design a Differentially-Private Faithful Federated Learning (DP-FFL) mechanism that is scalable while maintaining VCG's economic properties. In addition, our DP-FFL mechanism enables one to make three-way performance tradeoffs among privacy, convergence, and payment accuracy loss. *To the best of our knowledge, this is the first differentially private and faithful mechanism.*

II. Literature Review

Federated Learning. The existing literature has studied how to make the model sharing process more privacy-preserving (e.g., [6]–[9]), more secure (e.g., [10], [11]), more efficient (e.g., [12], [13]), and more robust (e.g., [14], [15]) against heterogeneity in the distributed data sources among many other works. For a more detailed survey, please refer to [16]. On the other hand, extensive studies attempting to improve the efficiency can be categorized into two directions: *algorithmic* and *communication* design. First, by designing new techniques including quantization (e.g., [12]) and new novel learning models (e.g., multi-task federated learning [13]). Second, in wireless edge networks, efforts have studied resource allocation algorithms for edge nodes (e.g., [18]), scheduling policies against interference (e.g., [19], [25]), mobile user selection and resource allocation algorithms (e.g., [21], [22]) and new communication techniques (e.g., over-the-air computation [23]). *However, this line of work assumes that*

[1]Specifically, Roberts' theorem states that, under mild conditions, the only incentive compatible mechanisms are VCG variants, which requires the revelation of agents' private information of their loss functions [41].

[2]We note that designing federated learning algorithms with state-of-the-art performances (e.g., convergence speed, cost-effectiveness, privacy, and robustness) is beyond the scope of this work.

agents (in addition to malicious attackers as in [10], [11]) are willing to participate into federated learning and obey the algorithms, whereas agents in practice are strategic and require proper incentives to do so.

In terms of incentive design for federated learning, which has been listed as an outstanding problem in [31], only a few recent studies attempted to address this issue [27]–[40]. Reference [37] describes a payoff sharing algorithm that maximizes the system designer's utility without considering agents' strategic behaviors. Yu *et al.* in [38] introduced fairness guarantees to the previous reward system. Other studies have been considering economic approaches to compensating agents' communication and computation costs based on economic approaches such as contract theory (e.g., [28], [32], [39]), Stackelberg game [33], [36], auction theory (e.g., [29], [34]), and reputation [32]. *However, this line of work focused on incentivizing agent participation by compensating them for their costs and eliciting their truthful cost information, but assumed that agents are obedient to follow federated learning algorithms without strategic manipulation.*

Faithful Mechanisms. Only a few studies in the literature considered faithful mechanism design (e.g., [42]–[45]). Faithful implementation was first introduced by Parkes *et al.* in [42]: a mechanism is faithful if no one can benefit from deviating, including information revelation, computation, and message passing. Feigenbaum *et al.* proposed a faithful policy-based inter-domain routing in [43]. Petcu *et al.* in [44] generalized the above results and achieve faithfulness for general distributed constrained optimization problems. *However, none of the existing studies on faithful implementation guaranteed differential privacy or scalability, or performed risk bound analysis in a (statistical) learning framework.*

III. System Model and Problem Formulation

A. System Overview

In this section, we introduce our federated learning model which aims to fit a global model over data that resides on, and has been generated by, a set $\mathcal{K} \triangleq \{k : 1 \leq k \leq K\}$ of agents (with distributed edge devices). The model also consists of a trusted (parameter) server. Each agent (e.g., a mobile device) has access to a local dataset $\mathcal{D}_k = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_k}$, where $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d$, $y_i \in \mathcal{Y} \subset \mathbb{R}$, $n_k = |\mathcal{D}_k|$ is the number of agent k's data samples, and $|\cdot|$ denotes the cardinality of a set. Sets $\mathcal{X}$ and $\mathcal{Y}$ are compact. We use $n \triangleq \sum_{k\in\mathcal{K}} n_k$ to denote the total number of data samples and $\mathcal{D} \triangleq \bigcup_{k\in\mathcal{K}} \mathcal{D}_k$ to denote the global dataset. The training data across the agents are often non-i.i.d., since the data of a given client is typically based on the usage of the particular edge device and may not be representative of the population distribution (e.g., [3]). To model the non-i.i.d. nature of the data, we assume that, every agent $k \in \mathcal{K}$ generates data via a distinct distribution $P_k(\boldsymbol{x}, y)$.

B. Federated Learning Setup

1) Expected risk: Ideally, a federated learning problem fits a global model $\boldsymbol{w} \in \mathbb{R}^d$ via *expected risk minimization*, i.e., by minimizing the following (weighted average) expected risk:

$$E(\boldsymbol{w}) = \sum_{k\in\mathcal{K}} p_k E_k(\boldsymbol{w}), \tag{1}$$

where $p_k \geq 0$ represents the weight for each agent k, satisfying $\sum_{k\in\mathcal{K}} p_k = 1$,[3] and $E_k(\boldsymbol{w})$ is agent k's local expected risk:

$$E_k(\boldsymbol{w}) \triangleq \int \ell(\boldsymbol{w}, \boldsymbol{x}, y) dP_k(\boldsymbol{x}, y), \ \forall k \in \mathcal{K}, \tag{2}$$

where $\ell(\cdot,\cdot,\cdot)$ is a per-sample loss function dependent on the model $\boldsymbol{w}$ applied to the input $\boldsymbol{x}_i$ and the label y_i.

2) Agent Modeling: The non-i.i.d. nature of data implies that agents have different $P_k(\cdot)$ and hence may have heterogeneous prediction objectives $E_k(\boldsymbol{w})$ and different preferences over the prediction outcome $\boldsymbol{w}$. Note that as the first work considering agent strategic manipulation in federated learning due to heterogeneous prediction objectives, we disregard the impact of resource consumption incurred in federated learning, which was considered in [32]–[36], [38]–[40].

Note that, under traditional mechanisms that do not account for federated learning (e.g., [41]–[45]), agents were assumed to know their exact objectives before participating. However, this is not the case here since their data distributions are unknown to themselves. Therefore, we assume that agents make their decisions based on probably approximate properties (instead of the deterministic ones in [41]–[45]) of the mechanisms to be formally introduced later.

3) Empirical risk: Each agent's local expected risk E_k is, however, not directly accessible since $P_k(\cdot,\cdot)$ is unknown. To solve (1) approximately, the induction principle of empirical risk minimization suggests to optimize an objective that averages the loss function on the training sets $\{\mathcal{D}_k\}_{k\in\mathcal{K}}$ instead [46]. Mathematically, federated learning algorithms aim to solve the following (*Empirical Risk Minimization (ERM)*) problem [3]:

$$\text{FL}: \quad \min_{\boldsymbol{w}} \ F(\boldsymbol{w}) \triangleq \sum_{k\in\mathcal{K}} p_k F_k(\boldsymbol{w}), \tag{3}$$

where $F_k(\boldsymbol{w})$ is the agent k's *local empirical risk (loss)*, given by

$$F_k(\boldsymbol{w}) = \frac{1}{n_k} \sum_{i=1}^{n_k} \ell(\boldsymbol{w}, \boldsymbol{x}_i, y_i), \ \forall k \in \mathcal{K}. \tag{4}$$

Let $\boldsymbol{w}^o$ denote the optimal solution to (3). One can anticipate that the optimal solution $\boldsymbol{w}^o$ to (3) approximates the solution to (1).[4] To quantify such a risk bound, we first adopt the following standard assumptions on the per-sample loss function $\ell(\cdot,\cdot,\cdot)$ throughout this paper (as in, e.g., [8], [9], [47]):

Assumption 1 (L_g-Smoothness)**.** *The gradients of the per-sample loss function $\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w}, \boldsymbol{x}, y)$ are well-defined and continuous such that there exists a constant L_g satisfying*

$$\|\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w}_1, \boldsymbol{x}, y) - \nabla_{\boldsymbol{w}}\ell(\boldsymbol{w}_2, \boldsymbol{x}, y)\|_2 \leq L_g \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2, \tag{5}$$

[3] As an example, **FedAvg** in [3] selects $p_k = n_k/n$ for all $k \in \mathcal{K}$.

[4] Throughout this work, we use empirical risk functions (e.g., $\{F_k\}$) in the objectives of federated learning problems and agents' payments, while we use expected risk functions (e.g., $\{E_k\}$) for risk bound analysis.

for any $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^d$ *and* $(\boldsymbol{x}, y) \in (\mathcal{X}, \mathcal{Y})$.

Assumption 2 (μ-Strong Convexity)**.** *The per-sample loss function* $\ell(\boldsymbol{w}, \boldsymbol{x}, y)$ *is* μ*-strongly convex in* $\boldsymbol{w}$ *for all* $(\boldsymbol{x}, y) \in (\mathcal{X}, \mathcal{Y})$*, i.e., for any* $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^d, (\boldsymbol{x}, y) \in (\mathcal{X}, \mathcal{Y})$*,*

$$\ell(\boldsymbol{w}_2, \boldsymbol{x}, y) \geq \ \ell(\boldsymbol{w}_1, \boldsymbol{x}, y) + \nabla \ell(\boldsymbol{w}_1, \boldsymbol{x}, y)^T (\boldsymbol{w}_2 - \boldsymbol{w}_1) + \frac{\mu}{2} \left\| \boldsymbol{w}_2 - \boldsymbol{w}_1 \right\|_2^2. \tag{6}$$

Typical examples that satisfy these assumptions include ridge regression, $l2$-norm regularized logistic regression, and softmax classifiers. We further use L_ℓ denote the maximal norm of the gradient of the per-sample loss at $\boldsymbol{w}^o$:

$$L_\ell \triangleq \max_{(\boldsymbol{x}, y) \in (\mathcal{X}, \mathcal{Y})} \nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}^o, \boldsymbol{x}, y). \tag{7}$$

We can now characterize the risk bound in the following:

Proposition 1. *The following inequality is true with a probability of* $1 - \delta$*:*

$$E(\boldsymbol{w}^o) - \min_{\boldsymbol{w}} E(\boldsymbol{w}) \leq \sum_{k \in \mathcal{K}} \frac{p_k^2}{n_k} \frac{L_\ell^2 d \log(2d/\delta)}{4\mu}, \tag{8}$$

where $\boldsymbol{w}^o$ *is the optimal solution to the federated learning problem in* (3)*,* d *is the dimension of* $\boldsymbol{w}$*.*

Due to the space limit, we present the proofs of all propositions, lemmas, theorems, and corollaries in [56]. The proof of Proposition 1 involves bounding $\nabla_{\boldsymbol{w}} E(\boldsymbol{w}^o)$ by the Hoeffding's inequality and leveraging the strong convexity of $E(\cdot)$. In the case of the **FedAvg** algorithm in [3], which selects $p_k = n_k/n$ for all $k \in \mathcal{K}$, the right hand side of (8) becomes $\frac{L_\ell^2 d \log(2d/\delta)}{4\mu n}$, which implies that the bound in this case converges to 0 as $n \to \infty$. The rate $\mathcal{O}(1/n)$ and is comparable to the result in [47], while we do not assume boundedness on $\boldsymbol{w}$ as [47] did.

C. Goals

The federated learning problem in (3) can be solved efficiently in a centralized manner if the server has the access to the global dataset $\mathcal{D}$, which is, however, impractical in the federated learning setting. Hence, as we mentioned in Section I, the focus of federated learning algorithms is on distributed learning that achieves *privacy preservation* and *scalability*. Moreover, here we also seek to ensure that agents will not opt out or tamper with the system. This requires a joint design of a federated learning algorithm and a proper (economic) mechanism. Specifically, a mechanism aims to achieve the following economic properties:

- (E1) *Efficiency*: The optimal solution (or its approximation, e.g., in Proposition 1) to (1) is achieved.
- (E2) *Faithful Implementation*[5]: Every agent does not have the incentive to deviate from the suggested federated learning algorithm.
- (E3) *Voluntary Participation*: Every agent should not be worse off by participating into the mechanism.
- (E4) (Weak) *Budget Balance (BB)*: The total payment from agents to the server is non-negative, i.e., the server is not required to inject money into the system.

As we have mentioned, we aim to achieve properties (E1)-(E4) in a probably approximate (but not deterministic) manner, due to the unavailability of data distributions $\{P_k\}_{k \in \mathcal{K}}$.

D. Mechanism Design

In order to achieve the above properties (E1)-(E4), the server designs an (economic) mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{S}^m, \boldsymbol{w}^*, \mathcal{P})$, described in the following:

- *Strategy space* $\mathcal{A} = \prod_{k \in \mathcal{K}} \mathcal{A}_k$: each agent can select a strategy $A_k \in \mathcal{A}_k$, representing the messages (potential misreports) to be submitted to the server in each iteration of a federated learning algorithm (such as gradient reporting in **FedAvg** [3]);
- *(Suggested) protocol/algorithm* $\mathcal{S}^m = \{s_k^m\}_{k \in \mathcal{K}}$: the server would like every agent to follow $\mathcal{S}^m$ by playing $A_k = s_k^m$ (e.g., agents' true gradients in each iteration in **FedAvg** [3]);
- *Learning updates* $\boldsymbol{w}^* : \mathcal{A} \to \mathbb{R}^d$ describes how the algorithm updates the model $\boldsymbol{w}$ in each iteration, which depends on agents' strategies $\boldsymbol{A}$;
- *Payment rule* $\mathcal{P} = \{\mathcal{P}_k\}_{k \in \mathcal{K}} : \mathcal{A} \to \mathbb{R}^K$ describes the payment each agent $k \in \mathcal{K}$ needs to pay to the server, depending on agents' strategies $\boldsymbol{A}$.

For a given mechanism $\mathcal{M}$, each agent k aims at minimizing its empirical cost, defined next.

Definition 1 (Overall loss)**.** *Each agent* k *has a (quasi-linear) overall loss, defined as*

$$J_k(A_k, \boldsymbol{A}_{-k}) = \mathcal{P}_k(\boldsymbol{A}) + F_k(\boldsymbol{w}^*(\boldsymbol{A})), \ \forall k \in \mathcal{K}. \tag{9}$$

In (9), we assume that each agent's objective is quasi-linear in its monetary loss, which is a standard assumption in economics [41].

There are two classical choices of the payment rules in the literature:

- The (weighted) VCG payment for each agent $k \in \mathcal{K}$ [41]:

$$\mathcal{P}_k^{\mathrm{VCG}} = \frac{1}{p_k} \left[\sum_{j \neq k} p_j F_j(\boldsymbol{w}^o) - \min_{\boldsymbol{w}} \sum_{j \neq k} p_j F_j(\boldsymbol{w}) \right]. \tag{10}$$

  The VCG mechanism is known as a generic truthful mechanism for achieving a socially-optimal solution (E1) and satisfies incentive compatibility and (E3) and (E4). Intuitively, the first term in (10) serves to align each agent's objective to the server's so that each agent also aims to minimize the global loss; the second term ensures that each agent does not overpay so as to ensure voluntary participation (E3). Detailed analysis of the VCG mechanism can be found in [41]. Nevertheless, as we have mentioned, the VCG payment cannot be directly applied here due to the communication/computation overhead. (3). Moreover, it assumes that agents know their exact objectives, which is not true here due to the unavailability

[5]Faithful implementation is also known as *incentive compatibility* or *strategyproofness*.

of $P_k(\cdot)$. Hence, in this paper, we will design distributed algorithms and corresponding mechanisms that lead to an approximate VCG payment in (10), to address the above issues and attain (E2) and (E3).

- Another possible payment rule is the Shapley value, which achieves other important properties in cooperative game theory [30], [48], but will not be addressed in this paper.

IV. Federated Learning Failure due to Strategic Manipulation

In this section, to illustrate the fact that agents to have incentives to opt out of the federated learning framework and manipulate the algorithms, we will demonstrate how non-i.i.d. and unbalanced data incentivizes strategic agents' misbehaviors. Specifically, we will analyze the conditions under which a pure federated learning algorithm (i.e. the one that does not leverage any economic mechanism) for optimizing (3) does not satisfy (E2) and (E3).

A. Why May Agents Prefer Local Learning?

In this subsection, we first understand why agents may rather not to participate into federated learning. In particular, for each agent $k \in \mathcal{K}$, we compare the achievable performances of federated learning (when all agents participate into and obey the algorithm, i.e., to solve (3))) and a *local learning algorithm* to independently solve the following local learning problem based on its local dataset $\mathcal{D}_k$:

$$\boldsymbol{w}_k^L \triangleq \arg\min_{\boldsymbol{w}} F_k(\boldsymbol{w}),\ \forall k \in \mathcal{K}. \tag{11}$$

In contrast, we use $\boldsymbol{w}^o$ to denote the optimal solution to the federated learning problem in (3):

$$\boldsymbol{w}^o \triangleq \arg\min_{\boldsymbol{w}} \sum_{k\in\mathcal{K}} p_k F_k(\boldsymbol{w}). \tag{12}$$

We start with the following corollary to understand the performance of local learning:

Corollary 1. *The following inequality is true with a probability of* $1-\delta$*:*

$$E_k(\boldsymbol{w}_k^L) - \min_{\boldsymbol{w}} E_k(\boldsymbol{w}) \leq \frac{L_\ell^2 d \log(2d/\delta)}{4\mu n_k}, \forall k \in \mathcal{K}. \tag{13}$$

The result in Corollary 1 is interpreted as a special case of Proposition 1.

We next introduce the following result that compares local learning and federated learning:

Proposition 2. *With a probability of* $1-\delta$*, federated learning in* (3) *leads to a risk bound of: for every agent* $k \in \mathcal{K}$*,*

$$E_k(\boldsymbol{w}^o) - \min_{\boldsymbol{w}} E_k(\boldsymbol{w}) \leq \sum_{j\in\mathcal{K}} \frac{p_j^2}{n_j} \frac{L_\ell^2 d \log(2d/\delta)}{4\mu} + 2\left\| P_k(\cdot) - \sum_{j\in\mathcal{K}} p_j P_j(\cdot) \right\|, \tag{14}$$

where $\|f(\cdot)\| \triangleq \int_t |f(t)| dt$*.*

The proof of Proposition 2 uses the envelope theorem [49].

The first term in the right-hand side of (14) characterizes the estimation error. The second term stands for the distance between its *data distribution* and the weighted average data distribution $\sum_{j\in\mathcal{K}} p_j P_j(\cdot)$, which characterizes how "typical" agent k is; a larger value of the second term implies that agent k is less typical. Intuitively, agent k being more typical implies that training samples of other agents are more useful in solving the agent k's prediction problem and hence resulting in a smaller bound in (14).

We note that the bounds in (13) and (14) are upper bounds that do not show which of the actual (expected) risks is worse. However, since agents do not know their exact data distributions but may estimate how typical they are based on some type of side information, they may rely on comparing these upper bounds in (13) and (14) to decide whether to participate into federated learning. Specifically, suppose that $p_k = n_k/n$ for all $k \in \mathcal{K}$ so that $1/n = \sum_{j\in\mathcal{K}} p_j^2/n_j$. For an agent with many samples so that n_k/n is close to one, then the first term in (14) will be close to the term in (13). Moreover, in such a case if the second term is large enough (i.e. the data is less typical), then this risk bound will be larger than that in (13). On the other hand, if an agent k has only a few samples so that n_k/n is small and the second term in (14) is small enough (i.e., its data is typical), then this risk bound in (14) will be smaller than that in (13). Collectively, we make an important observation from Corollary 1 and Proposition 2:

Remark 1. *The classical federated learning framework in* (3) *disincentivizes non-typical agents with sufficiently large datasets (such that* $1/n_k$ *is close to* $\sum_{j\in\mathcal{K}} p_j^2/n_j$*).*

B. Why May Agents Be Untruthful?

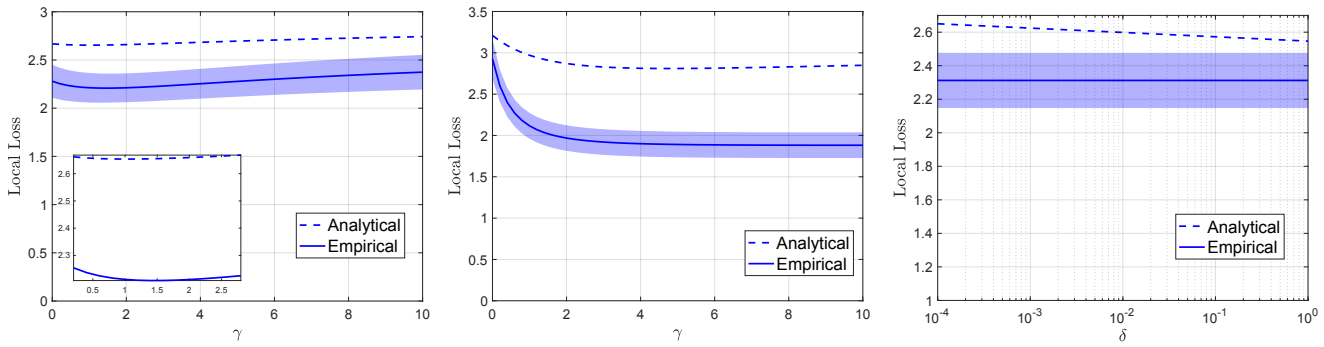
We next understand the incentive for agents to not follow a suggested federated learning algorithm even if they choose to participate.

Consider a heuristic (manipulation) strategy for agent k to amplify its reports (e.g., its gradients in **FedAvg** [3]) by a constant $\gamma > 1$, in each iteration of a federated learning algorithm. When agents other than k are obedient, the resultant manipulated federated learning algorithm is equivalent to solving the following problem:

$$\boldsymbol{w}_{k,\gamma} = \arg\min_{\boldsymbol{w}} \left(\sum_{j\neq k} p_j F_j(\boldsymbol{w}) + \gamma p_k F_k(\boldsymbol{w}) \right). \tag{15}$$

We can derive a risk bound for such a manipulation:

Proposition 3. *Suppose that agent* k *amplifies its report of its gradient by a constant coefficient* γ *and agents other than* k *report their truthful gradients for* ***FedAvg****. With a probability*



(a) (b) (c)

Fig. 1: An illustrative example of Proposition 3. In (a) and (b), we set mean $= 0.1$ and 2, respectively, fix $\delta = 0.01$, and compare the incurred local loss at different γ; in (c), we fix $\gamma = 3$ and mean $= 0.1$ and compare the incurred local loss at different δ.

of $1-\delta$*, the following inequality holds:*

$$E_k(\boldsymbol{w}_{k,\gamma}) - \min_{\boldsymbol{w}} E_k(\boldsymbol{w}) \tag{16}$$
$$\leq \frac{1}{(1+(\gamma-1)p_k)^2}\left(\sum_{j\neq k}\frac{p_j^2}{n_j}+\frac{\gamma^2 p_k^2}{n_k}\right)\frac{L_\ell^2 d\log(2d/\delta)}{4\mu}$$
$$+2\left\|P_k(\cdot)-\frac{1}{1+(\gamma-1)p_k}\left(\sum_{j\neq k}p_jP_j(\cdot)+\gamma p_k P_k(\cdot)\right)\right\|.$$

Proposition 3 is a direct application of Proposition 2. Note that, (16) becomes exactly the same as in (13) as by letting γ approach ∞, whereas (16) becomes exactly the same as in (14) when $\gamma = 1$. This indicates that, as stated in Remark 1, non-typical agents with sufficiently large datasets can benefit from choosing a relatively large γ.[6] Further, since local learning and federated learning without manipulation can be regarded as the special cases of the manipulated federated learning in (15), tampering with federated learning renders more capability and incentives to manipulate the system outcomes, compared to opting out of federated learning.

We consider an illustrative example of Proposition 3 (and Remark 1), as descried in the following. We consider a federate learning framework for two agents with $n_1 = 50$ and $n_2 = 400$ data samples, respectively. For each data sample (x_i, y_i) in either dataset $\mathcal{D}_1$ and $\mathcal{D}_2$, $x_i \in \mathbb{R}$ is randomly generated from a uniform distribution over $[0, 1]$. Labels y_i satisfy $y_i = -2x_i + 1 + \kappa_i$, where κ_i follows normal distributions $\mathcal{N}(\text{mean}, 2)$ and $\mathcal{N}(0, 2)$, truncated over $[-3, 3]$, for each data sample (x_i, y_i) generated in datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. Therefore, a mean closer to 0 implies that both agents' data distributions are closer. In Fig. 1, we study the impacts of the amplifying coefficient γ and δ on agent 1's empirical loss and its analytical (probably approximate) upper bound derived according to Proposition 3. We consider a scenario where two data distributions are close (as mean $= 0.1$) in Fig. 1(a). First, we observe that the agent 1's optimal coefficients γ to minimize its analytical and empirical local risks are both slightly larger than 1. On the other hand, when two data distributions are very distant (as mean $= 2$) as shown in Fig. 1(b), agent 1 prefers an infinitely large γ. Note that each agent's local loss as $\gamma \to \infty$ corresponds to its local loss under local learning. By comparing the values of local losses at $\gamma = 1$ and $\gamma = \infty$, Fig. 1 also validates our result in Remark 1: A non-typical agent prefers not to participate. Fig. 1(a)-(b) shows that analytical and empirical loss functions tend to have close minimizers. Finally, Fig. 1(c) demonstrates that δ only has a small impact on the tightness of the analytical bound since δ appears in a logarithmic function; $\delta = 0.01$ is already enough to ensure a reasonably tight analytical upper bound.

[6]It also implies that a strategic non-typical agent with a sufficiently large dataset may manipulate federated learning and lead to a system performance as worse as that of local learning.

C. When is Federated Learning Socially Efficient?

Even from the system-level perspective, with respect to minimizing the global expected risk in (1), federated learning may not be always more beneficial than local learning in (11), as we will analyze next.

With a probability of at least $1-\delta$, agents performing their respective local learning algorithms leads to an expected risk bound of

$$\sum_{k\in\mathcal{K}} p_k E_k(\boldsymbol{w}_k^L) - \sum_{k\in\mathcal{K}} p_k\left(\min_{\boldsymbol{w}} E_k(\boldsymbol{w})\right)$$
$$\leq \sum_{k\in\mathcal{K}}\frac{p_k}{n_k}\frac{L_\ell^2 d\log(2Kd/\delta)}{4\mu}. \tag{17}$$

Similarly, with a probability of at least $1-\delta$, federated learning in (3) leads to an expected risk bound of

$$E(\boldsymbol{w}^o) - \sum_{k\in\mathcal{K}} p_k\left(\min_{\boldsymbol{w}} E_k(\boldsymbol{w})\right) \leq \sum_{k\in\mathcal{K}}\frac{p_k^2}{n_k}\frac{L_\ell^2 d\log(2Kd/\delta)}{4\mu}$$
$$+\sum_{k\in\mathcal{K}} 2p_k\left\|P_k - \sum_{j\in\mathcal{K}} p_j P_j\right\|. \tag{18}$$

Consider a system with $n_k = n_j$ and $p_k = p_j$ for all k and j. Subtracting the risk bound of local learning from that of

federated learning yields

$$\frac{(K-1)L_\ell^2 d\log(2Kd/\delta)}{4\mu n} - \frac{2}{K}\sum_{k\in\mathcal{K}}\left\|P_k - \sum_{j\in\mathcal{S}}\frac{P_j}{K}\right\|. \quad (19)$$

We observe that the first term in (19) always increases in K while the second term needs not to do so, which implies that the federated learning is more likely to have a smaller risk bound when there are many agents in the system.

To generalize the above results, we can further cluster agents into several disjoint clusters $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_L$, and let agents in each cluster perform *intra-cluster learning*, i.e., they solve

$$\boldsymbol{w}^{\mathcal{C}_l} \triangleq \arg\min_{\boldsymbol{w}} \sum_{k\in\mathcal{C}_l} p_k F_k(\boldsymbol{w}). \quad (20)$$

Let $\hat{\mathcal{C}}(k)$ denote the cluster that agent k belongs to, i.e., if $k \in \mathcal{C}_l$, then $\hat{\mathcal{C}}(k) = \mathcal{C}_l$. The risk bound of such clustering and intra-cluster learning is given by

$$\begin{aligned}
&\sum_{k\in\mathcal{K}} p_k E_k(\boldsymbol{w}^{\hat{\mathcal{S}}(k)}) - \sum_{k\in\mathcal{K}} p_k\left(\min_{\boldsymbol{w}} E_k(\boldsymbol{w})\right)\\
&\le \sum_{k\in\mathcal{K}} \frac{p_{j,\hat{\mathcal{C}}(k)}^2}{n_k}\frac{L_\ell^2 d\log(2Kd/\delta)}{4\mu} + \sum_{k\in\mathcal{K}} 2p_k\left\|P_k - \bar{P}_{\hat{\mathcal{C}}(k)}\right\|\\
&\triangleq \mathrm{RB}_{\mathcal{C}}, \quad (21)
\end{aligned}$$

where $\mathcal{C}$ denotes the set of all clusters, i.e., $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_L\}$, $p_{k,\mathcal{A}} = p_k / \sum_{j\in\mathcal{A}} p_j$, and $\bar{P}_{\mathcal{A}}(\cdot,\cdot) = \sum_{j\in\mathcal{A}} p_{j,\mathcal{A}} P_j(\cdot,\cdot)$, for all clusters $\mathcal{A}$ in $\mathcal{C}$.

It is possible to design a clustering algorithm, based on agents' non-i.i.d. data distributions, to minimize the risk bound, which is beyond the scope of this work. Instead, we focus on the scenario in which federated learning is more beneficial than any clustering $\mathcal{C}$ for the system, by adopting the following assumption:

Assumption 3. *The federated learning leads to a tighter risk bound than any possible intra-cluster federated learning, i.e.,*

$$\mathrm{RB}_{\{\mathcal{K}\}} \le \mathrm{RB}_{\mathcal{C}}, \ \forall \mathcal{C}. \quad (22)$$

Assumption 3 can be true in a *massively distributed* federated learning framework [3], in which the average number of samples per agents n/K is much smaller than K. Even under Assumption 3, agents may still benefit from opting out of and tampering with a federated learning algorithm, which motivates the faithful federated learning mechanisms to be discussed in the following sections.

V. Faithful Federated Learning

In this section, we apply mechanism design for **FedAvg** in [4] to achieve a scalable federated learning algorithm (with the associated mechanism) that satisfies (E1)-(E3) approximately and (E4) exactly. Similar techniques could also be applied to other federated learning algorithms.

Algorithm 1: Faithful Federated Learning

1 The server initializes a model $\boldsymbol{w}[0]$ and step sizes η_1 and η_2 for Phase I and Phase II;
`// Phase I: Federated learning phase`
2 **for** ***iterations*** $t \in \{0, 1, ..., T_1\}$ **do**
3 The server broadcasts the current model $\boldsymbol{w}[t]$ to all agents;
4 Each agent k computes and reports its gradient to the server $\nabla F_k(\boldsymbol{w}[t])$.;
5 The server updates the model
$$\boldsymbol{w}[t+1] = \boldsymbol{w}[t] - \eta_1 \sum_{k\in\mathcal{K}} p_k \nabla F_k(\boldsymbol{w}[t]). \quad (23)$$
6 **end**
7 ***Return*** the approximately optimal model: $\boldsymbol{w}^* \triangleq \boldsymbol{w}[T_1]$;
`// Phase II: Payment computation phase`
8 **for** ***agents*** $k \in \mathcal{K}$ **do**
9 The server initializes the model $\boldsymbol{w}_{-k}[0] = \boldsymbol{w}^*$;
10 **while** ***iterations*** $t \in \{0, 1, ..., T_2\}$ *or* $\frac{1}{2\mu}\left\|\sum_{j\neq k} p_j \nabla F_j(\boldsymbol{w}_{-k}[t])\right\|_2^2 > p_k\epsilon$ **do**
11 Set $t \to t+1$;
12 The server broadcasts the current model $\boldsymbol{w}_{-k}[t]$ to all agents excluding k;
13 Agents $j \neq k$ compute and report their gradients to the server $\nabla F_j(\boldsymbol{w}_{-k}[t])$;
14 The server updates the model and agent k's payment:
$$\boldsymbol{w}_{-k}[t+1] = \boldsymbol{w}_{-k}[t] - \eta_2 \sum_{j\neq k} p_j \nabla F_j(\boldsymbol{w}_{-k}[t]), \quad (24a)$$
$$\mathcal{P}_k[t+1] = \mathcal{P}_k[t] + (\boldsymbol{w}_{-k}[t+1] - \boldsymbol{w}_{-k}[t])^T \cdot \sum_{j\neq k}\frac{p_j}{p_k}\nabla F_j(\boldsymbol{w}_{-k}[t]). \quad (24b)$$
15
16 **end**
17 ***Return*** the payment for agent k: $\mathcal{P}_k^* \triangleq \mathcal{P}_k[T_2]$;
18 **end**

A. Algorithm and Mechanism Description

We present the **Faithful Federated Learning (FFL)** algorithm in Algorithm 1, consisting of two phases: a *federated learning phase* (lines 1-7) and a *payment computation phase* (lines 8-18). In the first phase (consisting of T_1 iterations), we present a gradient-based federated learning algorithm to (approximately) attain the globally optimal solution $\boldsymbol{w}^*$ to (3), similar to FedAvg in [3].[7] The second phase consists of K outer iterations, with each computing each agent's payment without the need of directly revealing agents' private local empirical risk functions. We note that $\frac{1}{2\mu}\left\|\sum_{j\neq k} p_j \nabla F_j(\boldsymbol{w}_{-k}[t])\right\|_2^2 > p_k\epsilon$ in line 10 is to ensure

[7]Throughout this paper, we use superscript o to denote the exact optimal solution, and $*$ to denote the solution output by algorithms.

the accurate computation of each agents payment. As we will show later, accurately computed payments incentivize strategic agents to faithfully follow the intended federated learning algorithm in the first phase.

Based on Algorithm 1, we introduce the FFL mechanism. The reporting of each agent's true gradient (lines 4 and 12) in each iteration corresponds to the intended algorithm (that the server would like agents to follow), whereas each agent k report $\nabla\tilde{F}_k(\boldsymbol{w}[t]) \in \mathbb{R}^d$ is a potential misreport of its gradient. Formally, we have:

Definition 2 (The FFL Mechanism)**.** *The FFL mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{S}^m, \boldsymbol{w}^*, \mathcal{P})$ is described as: for every agent $k \in \mathcal{K}$,*

$$s_k^m = \{\nabla F_k(\boldsymbol{w}[t])\}_{t\in\mathcal{T}_{-k}} \quad \text{and} \quad A_k = \left\{\nabla\tilde{F}_k(\boldsymbol{w}[t])\right\}_{t\in\mathcal{T}_{-k}}, \tag{25}$$

where $\mathcal{T}_{-k} = \bigcup_{j\in\{0\}\cup\mathcal{K}\setminus\{k\}}\{1 \le t \le T_2\}$. The output global model $\boldsymbol{w}^*$ and each agent k's payment $\mathcal{P}_k$ are determined in (23) *and* (24)*, respectively.*

In the following, we explain the intuition behind Algorithm 1 and the FFL mechanism. We first define

$$\boldsymbol{w}_{-k}^o \triangleq \arg\min_{\boldsymbol{w}} \sum_{j\neq k} p_j F_j(\boldsymbol{w}). \tag{26}$$

Following Algorithm 1, the final payment for each agent k can be approximately expressed by

$$\begin{aligned}\mathcal{P}_k^* &= \sum_{t=1}^{T_2}\sum_{j\neq k}\frac{p_j}{p_k}\nabla F_j(\boldsymbol{w}[t])^T(\boldsymbol{w}[t+1]-\boldsymbol{w}[t])\\ &\approx \sum_{j\neq k}\int_{\boldsymbol{w}_{-k}[0]}^{\boldsymbol{w}_{-k}[T_2]}\frac{p_j}{p_k}\nabla F_j(\boldsymbol{w})^T d\boldsymbol{w}\\ &\approx \sum_{j\neq k}\frac{p_j}{p_k}\left[F_j(\boldsymbol{w}^o) - F_j(\boldsymbol{w}_{-k}^o)\right],\ \forall k\in\mathcal{K}.\end{aligned} \tag{27}$$

Therefore, the payment rule defined in Algorithm 1 is a VCG-like payment rule (i.e., it approximates (10)). As we have discussed, it can align each agent's objective to the server's objective, and hence potentially satisfies properties (E1)-(E4).

B. Federated Learning Phase

With Assumptions 1-3, the gradient-based learning algorithm in the federated learning phase of Algorithm 1 leads to the following standard linear convergence result [50]:

Lemma 1. *In the federated learning phase of Algorithm 1, if we choose a constant step size such that $\eta_1 = 1/L_g$, the gradient descent has a linear convergence rate of*

$$F(\boldsymbol{w}^*) - F(\boldsymbol{w}^o) \le \left(1-\frac{\mu}{L_g}\right)^{T_1}(F(\boldsymbol{w}[0]) - F(\boldsymbol{w}^o)). \tag{28}$$

Lemma 1 along with Proposition 1 implies that we can achieve (E1) in a *probably approximate* manner, as shown in the following:

Proposition 4. *In the federated learning phase of Algorithm 1, if we choose a constant step size such that $\eta_1 \le 1/L_g$, the following risk bound is true with a probability of $1-\delta$:*

$$E(\boldsymbol{w}^*) - \min_{\boldsymbol{w}} E(\boldsymbol{w}) \le \Phi(\delta), \tag{29}$$

for any $\delta \in (0,1)$, where

$$\begin{aligned}\Phi(\delta) \triangleq &\sum_{k\in\mathcal{K}}\frac{p_k^2}{n_k}\frac{L_\ell^2 d\log(2d/\delta)}{2\mu}\\ &+\frac{2L_g}{\mu}\left(1-\frac{\mu}{L_g}\right)^{T_1}(F(\boldsymbol{w}[0]) - F(\boldsymbol{w}^o)),\end{aligned} \tag{30}$$

C. Payment Computation Phase

In this subsection, we analyze the payment computation phase. We start with defining an optimal solution set $\mathcal{W}^*$ for all possible $\{p_k\}$:

$$\mathcal{W}^* \triangleq \left\{\arg\min_{\boldsymbol{w}}\sum_{k\in\mathcal{K}} p_k F_k(\boldsymbol{w}) : \forall p_k \ge 0 \text{ and } \sum_{k\in\mathcal{K}} p_k = 1\right\}, \tag{31}$$

and introduce the following gradient bound:

Definition 3 (Gradient Bound)**.** *The gradient bound L_f is defined as, for each agent $k \in \mathcal{K}$,*

$$\|\nabla_{\boldsymbol{w}} F_k(\boldsymbol{w})\|_2 \le L_f, \tag{32}$$

for all $\boldsymbol{w} \in \{\boldsymbol{w}[t]\}_{t\in\{0,1,\ldots,T_1\}} \cup \{\boldsymbol{w}_{-k}[t]\}_{k\in\mathcal{K},t\in\{0,1,\ldots,T_2\}} \cup \mathcal{W}^*$, where $\{\boldsymbol{w}[t]\}_{t\in\{0,1,\ldots,T_1\}}$ and $\{\boldsymbol{w}_{-k}[t]\}_{k\in\mathcal{K},t\in\{0,1,\ldots,T_2\}}$ are described in Algorithm 1.

Note that such a gradient bound always exists as the set $\{\boldsymbol{w}[t]\}_{t\in\{0,1,\ldots,T_1\}} \cup \{\boldsymbol{w}_{-k}[t]\}_{k\in\mathcal{K},t\in\{0,1,\ldots,T_2\}} \cup \mathcal{W}^*$ is compact[8] so that there always exists a large enough upper bound for all values of $\|\nabla_{\boldsymbol{w}} F_k(\boldsymbol{w})\|_2$ taken over the set.

We now present a formal bound of the absolute difference of the payment $\mathcal{P}_k^*$ and the exact VCG payment in the following:

Proposition 5. *The payment accuracy loss (the absolute difference between $\mathcal{P}_k^*$ and the VCG payment $\mathcal{P}_k^{\mathrm{VCG}}$ in* (10)*) is bounded by:*

$$\left|\mathcal{P}_k^* - \mathcal{P}_k^{\mathrm{VCG}}\right| \le \frac{1-p_k}{p_k} L_g L_f^2 (T_2+1)\eta_2^2,\ \forall k\in\mathcal{K}. \tag{33}$$

Intuitively, as Proposition 5 indicates, $\mathcal{P}_k^*$ converges to the VCG payment in (10) as the step sizes converge to zeros, i.e., $\eta_2 \to 0$.

In the following, we study the iteration complexity of the FFL mechanism, starting with the following lemma:

Lemma 2. *With equal weights ($p_k = 1/K$ for all $k \in \mathcal{K}$), the (Euclidean) distance between $\boldsymbol{w}_{-k}^o$ and $\boldsymbol{w}^o$ satisfies*

$$\left\|\boldsymbol{w}_{-k}^o - \boldsymbol{w}^o\right\|_2 \le \frac{L_f}{\mu K},\ \forall k\in\mathcal{K}. \tag{34}$$

[8]By the maximum theorem and the strong convexity of $F_k(\boldsymbol{w})$, $\arg\min_{\boldsymbol{w}}\sum_{k\in\mathcal{K}} p_k F_k(\boldsymbol{w})$ is continuous in $\{p_k\}_{\in\mathcal{K}}$. Therefore, the compactness of the set of all $\{p_k\}_{\in\mathcal{K}}$ satisfying $p_k \ge 0$ and $\sum_{k\in\mathcal{K}} p_k = 1$ indicates the compactness of $\mathcal{W}^*$.

Lemma 2 implies that the distance of $\boldsymbol{w}^o_{-k}$ to $\boldsymbol{w}^o$ is inversely proportional to the total number of agents K. In the following throughout this paper, we choose equal weights $p_k = 1/K$ for all $k \in \mathcal{K}$. Let $\lceil \cdot \rceil$ be the ceiling operator such that $\lceil x \rceil$ is the smallest non-negative integer that is no less than x. Based on Lemma 2, we now have one of the main results of this work:

Theorem 1. *Set $\eta_1 = 1/L_g$, $\eta_2 = 1/(KL_g)$, and $T_1 \geq \frac{2\ln(KG/\theta)}{\ln((1-\mu/L_g)^{-1})}$, for any $\theta > 0$. Set the number of iterations $T_2 = \left\lceil \ln\left(\frac{(L_f+\theta\mu)^2 L_g}{\mu^2 K\epsilon}\right) / \ln\left(\frac{L_g}{L_g-\mu}\right)\right\rceil$, for any $K > 0$ and $\epsilon > 0$ such that $\left[\ln\left(\frac{(L_f+\theta\mu)^2 L_g}{\mu^2 K\epsilon}\right) / \ln\left(\frac{L_g}{L_g-\mu}\right), \frac{L_g\epsilon K}{2L_f^2}\right]$ is not empty, we have a bounded payment accuracy loss, satisfying*

$$|\mathcal{P}_k[T_2] - \mathcal{P}_k^{\mathrm{VCG}}| \leq \epsilon, \ \forall k \in \mathcal{K}. \tag{35}$$

and a constant time complexity of

$$KT_2 \leq \left(1 + \frac{(L_f+\theta\mu)^2 L_g}{2\mu^2 e\epsilon}\right) / \ln\left(\frac{L_g}{L_g-\mu}\right) = \mathcal{O}(1). \tag{36}$$

Note that, for every ϵ, there always exists a large enough K such that the interval

$$\left[\ln\left(\frac{(L_f+\theta\mu)^2 L_g}{\mu^2 K\epsilon}\right) / \ln\left(\frac{L_g}{L_g-\mu}\right), \frac{L_g\epsilon K}{2L_f^2}\right] \tag{37}$$

is not empty.

Theorem 1 implies that, by ensuring the number of iterations in Phase I to satisfy $T_1 = \mathcal{O}(\log(K))$, the time complexity of the payment computation phase in Algorithm 1 is in fact $KT_2 = \mathcal{O}(1)$ with respect to K. In particular, a sufficiently large K such that $K \geq (L_f+\theta\mu)^2 L_g/(2\mu^2\epsilon)$ ensures that $|\mathcal{P}_k[0] - \mathcal{P}_k^{\mathrm{VCG}}| \leq \epsilon$ for all $k \in \mathcal{K}$. In other words, when K is sufficiently large, we can set $T_2 = 0$, in which case Phase II of Algorithm 1 will end without any iterations. Intuitively, the gradient-based nature benefits significantly from the Euclidean distance (between the $\boldsymbol{w}^o_{-k}$ and $\boldsymbol{w}^o$) inversely proportional to K. On the other hand, since the per-iteration communication complexity is $\mathcal{O}(K)$, the overall communication complexity is also $\mathcal{O}(K)$.

D. Properties

In this subsection, we will show that the FFL mechanism satisfies (E2) and (E3) approximately and (E4) exactly. We first introduce the following definition of (probably approximate) faithfulness as to achieve (E2) approximately:

Definition 4 (Faithfulness)**.** *A mechanism $\mathcal{M}$ is $(\tilde{\epsilon}, \tilde{\delta})$-faithful if the following bound holds for all agents $k \in \mathcal{K}$,*

$$\Pr\left\{\mathbb{E}[J_k(s_k^m, \boldsymbol{s}_{-k}^m)] - \min_{A_k \in \mathcal{A}_k} \mathbb{E}[J_k(A_k, \boldsymbol{s}_{-k}^m)] \leq \tilde{\epsilon}\right\} \geq 1 - \tilde{\delta}, \tag{38}$$

where $J_k(\cdot)$ is the overall loss introduced in Definition 1.

That is, the $(\tilde{\epsilon}, \tilde{\delta})$-faithfulness suggests that, when all other agents are following the suggested protocol, the incentive for agent k to deviate from doing so is small with a high probability of $1 - \tilde{\delta}$. The bound $\tilde{\epsilon}$ may depend on $\tilde{\delta}$ and the number of agents' data samples $\{n_k\}_{k\in\mathcal{K}}$ and is anticipated to vanish as $n \to \infty$. It follows that:

Proposition 6 (Faithful Implementation)**.** *If we choose $\eta_1 = 1/L_g$, $\eta_2 = \frac{1}{KL_g}$, $T_1 \geq \frac{2\ln(KG/\theta)}{\ln((1-\mu/L_g)^{-1})}$, and $T_2 = \left\lceil \ln\left(\frac{(L_f+\theta\mu)^2 L_g}{\mu^2 K\epsilon}\right) / \ln\left(\frac{L_g}{L_g-\mu}\right)\right\rceil$ for any $\theta > 0$ and $\epsilon > 0$ such that $\left[\ln\left(\frac{(L_f+\theta\mu)^2 L_g}{\mu^2 K\epsilon}\right) / \ln\left(\frac{L_g}{L_g-\mu}\right), \frac{L_g\epsilon K}{2L_f^2}\right]$ is not empty, then the FFL mechanism is $\left(\tilde{\epsilon}, \tilde{\delta}\right)$-faithful, where $\tilde{\epsilon}$ satisfies*

$$\tilde{\epsilon} = 2\epsilon + K\Phi(\tilde{\delta}), \quad \forall \tilde{\delta} \in (0,1), \tag{39}$$

where $\Phi(\delta)$ is defined in (30)*.*

The proof of Proposition 6 is an application of Theorem 1 and Proposition 1. An interesting observation is that, different from (16), data distributions $\{P_k(\cdot)\}$ do not appear in (39). Thus, we remark that:

Remark 2. *The faithful implementation property achieved by the FFL mechanism is robust against non-i.i.d. data. This differs substantially from the classical federated learning settings, in which non-typical agents may have incentives to manipulate federated learning algorithms, as shown in Propositions 2 and 3.*

To show the voluntary participation property, we consider the following probabilistic inequalities. From (17), the following inequality holds with a probability of $1-\delta$, for all $k \in \mathcal{K}$,

$$\begin{aligned}
&E_k(\boldsymbol{w}_k^L) - \min_{\boldsymbol{w}} E_k(\boldsymbol{w}) \\
\leq &\left(\sum_{j\neq k} \frac{p_j^2}{n_j(1-p_k)^2} + \frac{p_k}{n_k}\right) \frac{L_\ell^2 d\log(2Kd/\delta)}{4\mu} + \sum_{j\neq k} \frac{p_j}{p_k} \min_{\boldsymbol{w}} E_j(\boldsymbol{w}) \\
&- \min_{\boldsymbol{w}} \sum_{j\neq k} \frac{p_j}{p_k} E_j(\boldsymbol{w}) + 2\sum_{j\in\mathcal{K}} p_j \left\|P_j - \bar{P}_{\mathcal{K}\setminus\{k\}}\right\| \triangleq \mathrm{RB}_{k,\delta}^L.
\end{aligned} \tag{40}$$

Following (21), the following inequality holds with a probability of $1-\delta$, we have that the difference between each agent k's expected overall loss from participation into the FFL mechanism and $\min_{\boldsymbol{w}} E_k(\boldsymbol{w})$ satisfies

$$\begin{aligned}
&\mathbb{E}[J_k(\boldsymbol{s}_k^m, \boldsymbol{s}_{-k}^m)] - \min_{\boldsymbol{w}} E_k(\boldsymbol{w}) \\
\leq &\sum_{j\in\mathcal{K}} \frac{p_j^2}{n_j p_k} \frac{L_\ell^2 d\log(2Kd/\delta)}{4\mu} + \sum_{j\neq k} \frac{p_j}{p_k} \min_{\boldsymbol{w}} E_j(\boldsymbol{w}) \\
&- \min_{\boldsymbol{w}} \sum_{j\neq k} \frac{p_j}{p_k} E_j(\boldsymbol{w}) + 2\sum_{k\in\mathcal{K}} p_k \left\|P_k - \bar{P}_{\mathcal{K}}\right\| \triangleq \mathrm{RB}_{k,\delta}^{FFL}.
\end{aligned} \tag{41}$$

Based on the above probabilistic bounds and Assumption 3, we can derive the following result:

Proposition 7 (Risk-Bound-Based Voluntary Participation)**.** *If we choose $\eta_1 = 1/L_g$, $\eta_2 = 1/(KL_g)$, $T_1 \geq \frac{2\ln(KG/\theta)}{\ln((1-\mu/L_g)^{-1})}$, and $T_2 = \left\lceil \ln\left(\frac{(L_f+\theta\mu)^2 L_g}{\mu^2 K\epsilon}\right) / \ln\left(\frac{L_g}{L_g-\mu}\right)\right\rceil$ for any $\theta > 0$ $\epsilon > 0$ such that $\left[\ln\left(\frac{(L_f+\theta\mu)^2 L_g}{\mu^2 K\epsilon}\right) / \ln\left(\frac{L_g}{L_g-\mu}\right), \frac{L_g\epsilon K}{2L_f^2}\right]$ is not empty, then the following inequality holds, for all $k \in \mathcal{K}$,*

$$\mathrm{RB}_{k,\delta}^{FFL} \leq \mathrm{RB}_{k,\delta}^L + \epsilon + K\Phi(\tilde{\delta}). \tag{42}$$

Intuitively, although agents do not know their exact data distributions, Proposition 7 suggests that the risk bound of the FFL mechanism is smaller than that of local learning plus ϵ. This incentivizes agents to voluntarily participate into the FFL mechanism, achieving (E3).

Finally, we show that the FFL mechanism satisfies (E4) in the following:

Proposition 8 (Budget Balance)**.** *The FFL mechanism $\mathcal{M}$ achieves budget balance (E4):*

$$\sum_{k\in\mathcal{K}} \mathcal{P}_k^* \geq 0. \tag{43}$$

To summarize, our FFL algorithm and the FFL mechanism achieve all the desired economic properties of (E1)-(E4). In addition, our FFL mechanism is also scalable as it only incurs an iteration complexity of $\mathcal{O}(\log(K))$ preserves agent privacy as it does not directly require agents to reveal their empirical risks or training data.

VI. Differentially Private Faithful Federated Learning

In this section, we aim to design a faithful federated learning mechanism achieving a more rigorous guarantee of privacy. We design a scalable VCG payment, and leverage differential privacy and secure multi-party computation to design a differentially private faithful federated learning algorithm and the corresponding mechanism.

A. Scalable VCG Payment

The VCG payment in (10) for a system with K agents requires one to solve $K+1$ optimization problems, which incurs considerable communications and computation overheads for a large-scale system. We showed in Section V that the time complexity of solving these problems using a gradient-based algorithm is $\mathcal{O}(\log(K))$ with respect to K. However, to achieve differential privacy, as we will show next, one relies on gradient perturbation under which the number of iterations for each problem no longer decreases in K. To this end, we introduce a scalable approximation of the VCG payment in (10) by reducing the number of problems to be solved.

We formally introduce the Scalable VCG payment in the following:

Definition 5 (Scalable VCG Payment)**.** *We randomly divide the set of agents $\mathcal{K}$ into $\mathcal{L} = \{1, 2, ..., L\}$ disjoint clusters. Each cluster is indexed by l and denoted by $\mathcal{C}_l$. We properly divide $\mathcal{K}$ in such a way that each cluster $\mathcal{C}_l$ has either $\lceil \frac{K}{L} \rceil$ or $\lfloor \frac{K}{L} \rfloor$ agents.*[9]

The scalable VCG payment for each agent k is

$$\mathcal{P}_k^{\mathrm{S}} = \frac{1}{p_k} \sum_{j\neq k} p_j \left(F_j(\boldsymbol{w}^o) - F_j(\boldsymbol{w}_l^o) \right), \ \forall k \in \mathcal{C}_l, \ l \in \mathcal{L}, \tag{44}$$

where

$$\boldsymbol{w}_l^o = \arg\min_{\boldsymbol{w}} \sum_{k\in\mathcal{K}/\mathcal{C}_l} p_k F_k(\boldsymbol{w}), \ \forall l \in \mathcal{L}. \tag{45}$$

Hence, we approximate $\boldsymbol{w}_{-k}^o$ for all $k \in \mathcal{C}_l$ by $\boldsymbol{w}_l^o$. In this case, instead of solving K optimization problems, we only need to solve L optimization problems. In the following theorem, we introduce a proper way to select L:

Theorem 2. *If we select*

$$L \geq \min\left\{ K, \sqrt{\frac{L_g(K-1)}{2\epsilon}} \frac{L_f}{\mu} \right\} = \mathcal{O}\left(\sqrt{\frac{K}{\epsilon}} \right), \tag{46}$$

then the Scalable VCG Payment in Definition 5 leads to an approximation error of

$$\left| \mathcal{P}_k^{\mathrm{S}} - \mathcal{P}_k^{\mathrm{VCG}} \right| \leq \epsilon, \ \forall k \in \mathcal{K}, \tag{47}$$

where $\mathcal{P}_k^{\mathrm{VCG}}$ is the VCG payment for agent k in (10)*.*

The proof of Theorem 2 uses a similar technique to that of Lemma 2, as $\left\| \boldsymbol{w}_{-k}^o - \boldsymbol{w}_l^o \right\|_2$ is inversely proportional to the number of agents within each cluster $\mathcal{C}_l$.

Theorem 2 indicates that, to maintain a bounded approximation error, the number of optimization problems to be solved grows at a square root rate, compared to the classical VCG mechanism with a linear rate. Therefore, the Scalable VCG payment in Definition 5 allows us to design a more scalable mechanism when we cannot rely on a gradient-based algorithm.

B. Differentially Private FFL Algorithm and Mechanism

Motivated by a recent differentially private federated learning algorithm in [8], we next introduce the techniques to ensure differential privacy by combining secure multi-party computation (to aggregate agents' local gradients) and gradient perturbation. Jayaraman *et al.* in [8] showed that this allows the server to add only a single noise copy, and can outperform the algorithms requiring local gradient perturbation before aggregation.

Before we describe and analyze the algorithm, we first formally introduce the following concepts:

1) Differential Privacy: We aim to guarantee *differential privacy*, which is a crypographically-motivated notion of privacy [51]. We define $\alpha \geq 0$ as privacy risk. Formally, we have:[10]

Definition 6 ((α, β)-differential privacy (DP))**.** *Let α be a positive real number and $\mathcal{Z}$ be a randomized algorithm. The algorithm $\mathcal{Z}$ is said to provide (α, β)-DP if, for any two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ that differ on a single element, $\mathcal{Z}$ satisfies*

$$\Pr[\mathcal{Z}(\mathcal{D}_1) = y] \leq \exp(\alpha) \cdot \Pr[\mathcal{Z}(\mathcal{D}_2) = y] + \beta, \ \forall y. \tag{51}$$

The above definition is reduced to the α-DP when $\beta = 0$, as in [52]. As we will show, one can achieve (α, β)-DP by adding noise sampled from Gaussian distributions to gradients.

[9] As an example, a set of $K = 18$ agents can be divided into the following 4 clusters: $\mathcal{C}_1 = \{1, 4, 5, 14\}, \mathcal{C}_2 = \{2, 3, 7, 10, 15\}, \mathcal{C}_3 = \{11, 13, 16, 17\}$, and $\mathcal{C}_4 = \{6, 8, 9, 12, 18\}$.

[10] Note that (α, β)-differential privacy is also known as (ϵ, δ)-differential privacy, as in [8].

Algorithm 2: Differentially Private Faithful Federated Learning (DP-FFL)

1 The server initializes $\boldsymbol{w}[0]$ and step sizes η_1, η_2;
// Phase I: Federated learning phase
2 **for** ***iterations*** $t \in \{0, 1, ..., T_1\}$ **do**
3 The server broadcasts $\boldsymbol{w}[t]$ to all agents;
4 Each agent k computes and reports its gradient to the server $\nabla F_k(\boldsymbol{w}[t])$;
5 The server *securely aggregates agents' gradients, adds noise*, and updates the model according to:

$$\boldsymbol{w}[t+1] = \boldsymbol{w}[t] - \eta_1 \left(\sum_{k \in \mathcal{K}} p_k \nabla F_k(\boldsymbol{w}[t]) + \boldsymbol{n} \right), \tag{48}$$

where $\boldsymbol{n}$ is a random vector sampled from $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_d)$ and σ^2 is given in (53);
6 **end**
7 ***Return*** $\boldsymbol{w}^* = \boldsymbol{w}[T_1]$;
8 **for** ***clusters*** $l \in \mathcal{L}$ **do**
// Phase II: Payment computation phase
9 The server initializes the model $\boldsymbol{w}_l[0] = \boldsymbol{w}^*$;
10 **for** ***iterations*** $t \in \{0, 1, ..., T_2\}$ **do**
11 The server broadcasts $\boldsymbol{w}_l[t]$ to all agents not in $\mathcal{C}_l$;
12 Each agent $k \notin \mathcal{C}_l$ computes and reports its gradient to the server $\nabla F_k(\boldsymbol{w}_l[t])$;
13 The server *securely aggregates agents' gradients and adds noise* to update the model:

$$\boldsymbol{w}_l[t+1] = \boldsymbol{w}_l[t] - \eta_2 \left(\sum_{j \in \mathcal{C}_l} p_j \nabla F_j(\boldsymbol{w}_l[t]) + \boldsymbol{n} \right), \tag{49}$$

14 where $\boldsymbol{n}$ is a random vector sampled from $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_d)$ and σ^2 is given in (53);
15 **end**
16 Each agent $k \in \mathcal{K}$ reports the value $F_k(\boldsymbol{w}^*) - F_k(\boldsymbol{w}_l[T_2])$ and the server computes

$$\mathcal{P}_k^* = \sum_{j \neq k} \frac{p_j}{p_k} \left(F_j(\boldsymbol{w}^*) - F_j(\boldsymbol{w}_l[T_2]) \right) + n_{P,k}, \tag{50}$$

for all $k \in \mathcal{C}_l$, where $n_{P,k}$ is sampled from $\mathcal{N}(0, \sigma_P^2)$ and σ_P^2 is given in (53);
17 **Return** the payment $\mathcal{P}_k^*$ for agent $k \in \mathcal{C}_l$;
18 **end**

2) Secure Multi-Party Computation: For preserving the privacy of agents' inputs without revealing them to others, the server aims to securely aggregate their gradients in each iteration.[11] To this end, we consider secure multi-party computation (MPC) protocols that enable one to jointly aggregate their private inputs. Examples of these are protocols that employ cryptographic techniques (e.g., homomorphic encryption and secret sharing). In this work, we do not focus on improving or evaluating the MPC protocols, since the methods we propose can be implemented using standard MPC techniques.[12]

3) Algorithm and Mechanism Description: We are ready to introduce the **Differentially Private Faithful Federated Learning (DP-FFL) algorithm** in Algorithm 2, which also consists of two phases. Compared to the FFL algorithm in Algorithm 1, we add noise in lines 5, 14, and 16 based on

$$\sigma^2 = \frac{16 L_f^2 (T_1 + K T_2) \log(1/\beta)}{K^2 n_{(1)}^2 \alpha^2} \tag{52}$$

$$\sigma_P^2 = \frac{16 K \log(1/\beta)}{n_{(1)}^2 \alpha^2}, \tag{53}$$

where $n_{(1)}$ is the size of smallest local dataset among all agents $\mathcal{K}$. We also adopt the Scalable VCG payment from Definition 5 in the payment computation phase of Algorithm 2.

In the following, we introduce the DP-FFL mechanism associated to Algorithm 2:

Definition 7 (The DP-FFL Mechanism)**.** *The DP-FFL mechanism $\mathcal{M}^{\mathrm{P}} = (\mathcal{A}, \mathcal{S}^m, \boldsymbol{w}^*, \mathcal{P})$ satisfies, for all agents $k \in \mathcal{K}$,*

$$s_k^m = \{\nabla F_k(\boldsymbol{w}[t])\}_{t \in \mathcal{T}_{-k}} \quad \text{and} \quad A_k = \{\nabla \tilde{F}_k(\boldsymbol{w}[t])\}_{t \in \mathcal{T}_{-k}}, \tag{54}$$

where $\mathcal{T}_{-l} = \bigcup_{j \in \{0\} \cup \mathcal{L} \setminus \{l\}} \{1 \leq t \leq T_2\}$. The output global model $\boldsymbol{w}^*$ and each agent k's payment $\mathcal{P}_k$ are determined in (48) *and* (50)*, respectively.*

We re-define the gradient bound specific for Algorithm 2 in the following:

Definition 8 (Gradient Bound)**.** *The gradient bound L_f is defined as for each agent $k \in \mathcal{K}$,*[13]

$$\|\nabla_{\boldsymbol{w}} F_k(\boldsymbol{w})\|_2 \leq L_f, \tag{55}$$

for all $\boldsymbol{w} \in \{\boldsymbol{w}[t]\}_{t \in \{0,1,...,T_1\}} \cup \{\boldsymbol{w}_l[t]\}_{l \in \mathcal{L}, t \in \{0,1,...,T_2\}} \cup \mathcal{W}^*$, where $\{\boldsymbol{w}[t]\}_{t \in \{0,1,...,T_1\}} \cup \{\boldsymbol{w}_l[t]\}_{l \in \mathcal{L}, t \in \{0,1,...,T_2\}}$ are described in Algorithm 2 and $\mathcal{W}^*$ is defined in (31)*.*

We now show that DF-FFL achieves the following privacy property:

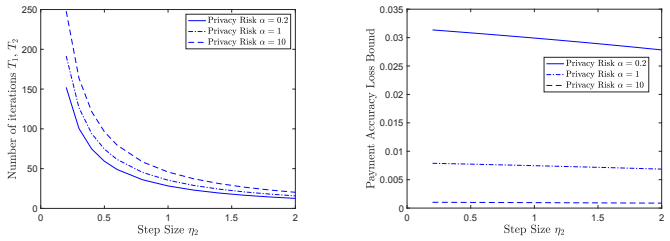
Proposition 9. *Algorithm 2 is (α, β)-differentially private.*

Proof of Proposition 9 is based on [8] and [54]. Similar to Proposition 4, we can also prove (E1) for the DF-FFL mechanism. We present the detailed analysis in [56].

[11] Note that MPC protocols are only able to protect the training data during the learning process, whereas the resulting model of a federated learning algorithm still relies on the differential privacy techniques (by adding noise in our proposed DP-FFL algorithm) against inferring private data of each agent.

[12] A concrete example of the standard MPC technique in federated learning frameworks can be found in [8], [9].

[13] For readability, we are overloading the notation L_f to avoid introducing additional parameter names.



(a) (b)

Fig. 2: Impacts of η_2 and α on the iterations needed T_1, T_2 and payment accuracy loss bounds $\left|\mathbb{E}[\mathcal{P}_k^*] - \mathcal{P}_k^{\mathrm{VCG}}\right|$.

C. Three-Way Tradeoffs between Privacy, Accuracy, and the Iterations Needed

We will discuss three-way performance tradeoffs in the following:

Proposition 10. *If we choose constant step sizes* $\eta_1 \leq 1/L_g$, $\eta_2 \leq \frac{1}{(K-1)L_g}$, $L \geq \min\left\{K, \sqrt{\frac{L_g(K-1)}{2\epsilon}}\frac{L_f}{\mu}\right\}$, *for any* $\epsilon > 0$, *and the iterations as*

$$T_1 = T_2 = \left\lceil \frac{\log\left(\frac{B\log(C^{-1})}{A}\right)}{\log(C^{-1})} \right\rceil, \tag{56}$$

then Algorithm 2 leads to a bounded expected payment accuracy loss:

$$\left|\mathbb{E}[\mathcal{P}_k^*] - \mathcal{P}_k^{\mathrm{VCG}}\right| \leq A\left(\frac{1+\log(\frac{B\log(C^{-1})}{A})}{\log(C^{-1})}\right) + \epsilon, \tag{57}$$

for all agents $k \in \mathcal{K}$, *where*

$$A = \frac{8\eta_2 d L_f^2 (K+1)\log(1/\beta)}{\mu K^2 (K-1) n_{(1)}^2 \alpha^2}, \tag{58a}$$

$$B = \frac{(K-1)L_f^2}{2\mu}, \tag{58b}$$

$$C = 1 - (K-1)\mu\eta_2. \tag{58c}$$

Proposition 10 suggests a three-way tradeoff among privacy (α, β), the iterations needed (T_1 and T_2), and accuracy $\left|\mathbb{E}[\mathcal{P}_k^*] - \mathcal{P}_k^{\mathrm{VCG}}\right|$. That is, by fixing β and properly selecting α and η_2, one can improve two performance metrics by trading off the third. To see this, we consider the following three scenarios as guidelines to make such a three-way tradeoff:

1) *To improve both privacy and iteration complexity by sacrificing accuracy*, one can set $\alpha \to 0$ to ensures privacy and set η_2 as a positive constant. In this case, we have that $A \to \infty$, and hence $T_1 = T_2 \to 0$. However, the payment accuracy loss in (57) diverges to infinity, i.e., $\left|\mathbb{E}[\mathcal{P}_k^*] - \mathcal{P}_k^{\mathrm{VCG}}\right| \to \infty$.
2) *To improve both accuracy and iteration complexity by sacrificing privacy*, one can set α and η_2 in such a way that η_2/α^2 is constant (and hence A is also constant). Further, by increasing η_2 (and hence increasing α as well), we have that C decreases, and hence $\log(\log(C^{-1}))/\log(C^{-1})$ decreases. Therefore, such a strategy decreases T_1, T_2, and $\left|\mathbb{E}[\mathcal{P}_k^*] - \mathcal{P}_k^{\mathrm{VCG}}\right|$ at the same time. However, η_2 is upper-bounded by the maximal step size $\frac{1}{(K-1)L_g}$.
3) *To achieve perfect accuracy* $\left|\mathbb{E}[\mathcal{P}_k^*] - \mathcal{P}_k^{\mathrm{VCG}}\right| \to 0$, *one needs to sacrifice both iteration complexity, privacy, and scalability.* Specifically, by setting $\alpha \to \infty$, it follows that $A \to 0$ and hence $T_1 = T_2 \to \infty$. Letting $L = K$ as well, we have $\left|\mathbb{E}[\mathcal{P}_k^*] - \mathcal{P}_k^{\mathrm{VCG}}\right| \to 0$.

We present a numerical example of Proposition 10 in Fig. 2, which compares the iterations needed and payment accuracy loss bounds at different α and η_2. An interesting observation is that the payment accuracy loss bound hardly changes when η_2 decreases. This results from the fact that $\log((1-(K-1)\mu\eta_2)^{-1}) \approx (K-1)\mu\eta_2$, which makes $A/\log(C^{-1})$ almost constant if we only tune η_2. On the other hand, fixing the step size η_2, T_1 and T_2 decrease in α and the payment accuracy loss bound increases in α.

D. Properties

Collectively, in this subsection, we will show that the DP-FFL algorithm (and the DP-FFL mechanism) satisfies (E2) and (E3) approximately and (E4) exactly.

Corollary 2 (Faithful Implementation)**.** *If we choose* $\eta_1 \leq 1/L_g$, $\eta_2 \leq \frac{1}{(K-1)L_g}$ $T_1 = T_2 = \left\lceil \log\left(\frac{B\log(C^{-1})}{A}\right)/\log(C^{-1})\right\rceil$, *and* $L \geq \min\left\{K, \sqrt{\frac{L_g(K-1)}{2\epsilon}}\frac{L_f}{\mu}\right\}$, *then the DP-FFL mechanism is* $(\tilde{\epsilon}, \tilde{\delta})$*-faithful, where*

$$\tilde{\epsilon} = 2\epsilon + 2A\left(\frac{1+\log(\frac{B\log(C^{-1})}{A})}{\log(C^{-1})}\right) + \sum_{k\in\mathcal{K}}\frac{1}{n_k}\frac{L_\ell^2 d\log(2d/\tilde{\delta})}{2K\mu} + D\frac{L_f^2 d\log(Kn_{(1)})\log(1/\beta)}{n_{(1)}^2\alpha^2}, \tag{59}$$

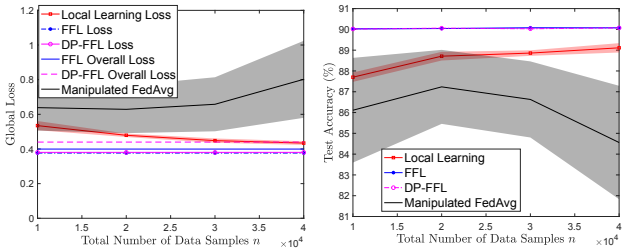
and A, B, *and* C *are defined in* (58)*, and* D *is some positive constant.*

Corollary 2 is a direct application of Propositions 6 and 10 and Theorem 2; the three terms on the right-hand side of (59) come from Theorem 2, Proposition 10, and Proposition 6, respectively.

Corollary 3 (Risk-Bound-Based Voluntary Participation)**.** *If we choose* $\eta_1 \leq \frac{1}{L_g}$, $\eta_2 \leq \frac{1}{(K-1)L_g}$ $T_1 = T_2 = \left\lceil \log\left(\frac{B\log(C^{-1})}{A}\right)/\log(C^{-1})\right\rceil$, *and* $L \geq \min\left\{K, \sqrt{\frac{L_g(K-1)}{2\epsilon}}\frac{L_f}{\mu}\right\}$, *then the DP-FFL algorithm and the DP-FFL mechanism lead to the following inequality:*

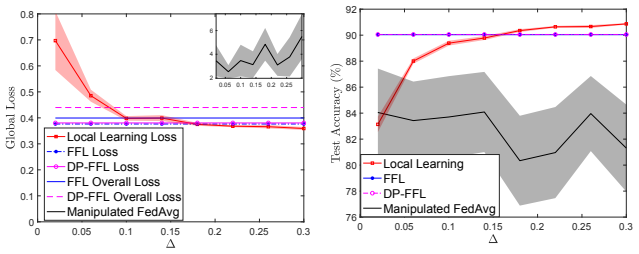
$$\mathrm{RB}_{k,\delta}^{FFL} \leq \mathrm{RB}_{k,\delta}^{L} + \epsilon + D\frac{L_f^2 d\log(Kn_{(1)})\log(1/\beta)}{n_{(1)}^2\alpha^2} + \sum_{k\in\mathcal{K}}\frac{1}{n_k}\frac{L_\ell^2 d\log(2d/\tilde{\delta})}{2K\mu} + A\left(\frac{1+\log(\frac{B\log(C^{-1})}{A})}{\log(C^{-1})}\right) \tag{60}$$

for all agents $k \in \mathcal{K}$, *where* $\mathrm{RB}_{k,\delta}^{L}$ *and* $\mathrm{RB}_{k,\delta}^{FFL}$ *are defined in* (40) *and* (41)*, respectively.*



(a) (b)

Fig. 3: Impacts of training data samples n on (a) training loss and overall costs and (b) test accuracy. We set $K = 10$, $\alpha = 0.1$, $\beta = 0.01$, $\Delta = 0.05$, $T_1 = 80$, and $T_2 = 20$.



(a) (b)

Fig. 4: Impacts of non-i.i.d. data, characterized by Δ, on (a) training loss and overall costs and (b) test accuracy. We set $K = 10$, $\alpha = 0.1$, $\beta = 0.01$, $n = 10000$, $T_1 = 80$, and $T_2 = 20$.

We note that, as implied in Corollary 3, performing local learning does not incur privacy loss for individual agents. Therefore, comparing the results in Corollary 3 and Proposition 7, the DP-FFL algorithm leads to a worse risk bound than the FFL algorithm.

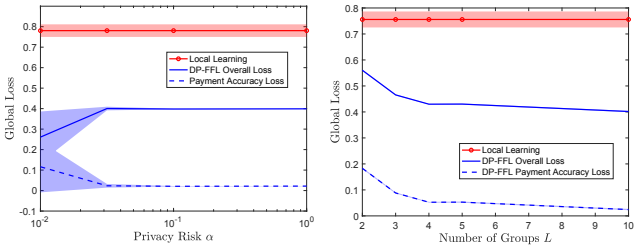
Finally, Proposition 10 also implies that the DP-FFL mechanism achieves budget balance (E4) approximately. We present the detailed analysis in [56].

VII. Evaluation

In this section, we evaluate our proposed FFL and DP-FFL mechanisms with $K = 10$ agents. We consider regularized multinomial logistic regression for the MNIST dataset with 60,000 training samples and 10,000 testing samples [55]. We uniformly randomly allocate samples with label y to all agents whose last digits of their indices are y. For each sample allocated to agents, with a probability of $1 - \Delta$, we reallocate this sample to a random agent with equal probabilities. Therefore, the degree of heterogeneity in this non-i.i.d. data can be characterized by Δ; a larger Δ leads to greater data heterogeneity and $\Delta = 0$.

For performance comparison, we compare our proposed FFL and DP-FFL schemes against two benchmarks: i) a gradient-based local learning benchmark, in which agents independently solve (11), and ii) a manipulated FedAvg benchmark, in which the server intends to execute FedAvg [3], while one agent manipulates the federated learning algorithm by multiplying its gradient report by an amplifying coefficient γ in each iteration. We compare the global loss, $F(\boldsymbol{w})$, and the weighted average test accuracy achieved by different schemes.

Impact of the total number of training samples: We study the impact of the number of total training data samples n in Fig. 3. First, we show that our proposed schemes significantly outperform the manipulated FedAvg, implying that federated learning manipulated even only by one agent can lead to significant performance loss. Therefore, it demonstrates the importance of faithful implementation of federated learning algorithms. Second, both proposed schemes outperform local learning with respect to either global loss and test accuracy. This also indicates that agents are willing to voluntarily participate in federated learning, as both proposed schemes achieve smaller overall losses. We observe that the total number of training data samples n has a greater impact on both local learning and the manipulated FedAvg benchmarks



(a) (b)

Fig. 5: Impacts of (a) α and (b) L. We set $K = 10$, $n = 10000$, $\beta = 0.01$, $\Delta = 0.05$, $T_1 = 80$, and $T_2 = 20$. We further set $L = 10$ in (a) and $\alpha = 0.1$ in (b).

than the proposed (FFL and DP-FFL) mechanisms. This is the performances both benchmarks are more sensitive to the sizes of local datasets, compared to federated learning.

Impact of non-i.i.d. data: We next study the impact of varying the heterogeneity in the data given by Δ in Fig. 4. As shown in Fig. 4, Δ only has an impact on local learning but not on the proposed mechanisms. In Fig. 4(a), we show that, in terms of each individual agent's objective (overall loss), the DP-FFL and the FFL mechanisms outperform local learning, when $\Delta \leq 0.15$. Local learning is more beneficial, compared to federated learning, when the agents have considerably non-i.i.d. data distributions (i.e., $\Delta \geq 0.15$). In particular, a higher degree of heterogeneous data implies each agent has a higher portion of (both training and test) data samples with labels corresponding to its own index (e.g., agent 3 may have more (both training and test) data samples with label 3 when Δ increases). This means that individual local datasets are more "useful" when Δ is large, therefore, incurring a higher test accuracy for local learning. Finally, Fig. 4 (a) and (b) imply whenever local learning is less beneficial than the proposed mechanisms regarding the test accuracy, the FFL and the DP-FFL schemes are more profitable for individual agents, which is consistent with our risk-bound-based voluntary participation results in Proposition 7 and Corollary 3 under Assumption 3.

Impacts of privacy risk α and the number of group L. We study the impacts of the privacy risk α and the number of groups L in Fig. 5. Note that the manipulated FedAvg and the proposed FFL algorithm are not directly comparable here, as they cannot guarantee differential privacy. We set $\Delta = 0.05$ and $K = 10$. Fig. 5 (a) shows that both the mean and the standard deviation of the payment accuracy loss decrease in α, as a larger privacy risk α leads to less noise in the DP-

FFL algorithm. An interesting observation is that, to attain a reasonably small payment accuracy loss, one should choose a small T_2 for a small α, which is consistent with Proposition 10. Fig. 5 (b) shows that increasing the number of groups L reduces the payment accuracy loss. In addition, a relatively large enough number of groups (i.e., $L \geq 4$) is enough to maintain a relative small payment accuracy loss.

VIII. Conclusions

We have studied an economic approach to federated learning robust against strategic agents' manipulation. We have analyzed how the key feature of federated learning, unbalanced and non-i.i.d. data, affects agents' incentive to voluntarily participate and obediently follow federated learning algorithms. We have designed the first faithful mechanism for federated learning, achieving (provably approximately) optimality, faithful implementation, voluntary participation, with the time complexity (in terms of the number of agents K) of $\mathcal{O}(\log K)$. We have further presented the differentially private faithful federated learning mechanism, which is the first differentially private faithful mechanism. It provides scalability, maintains the economic properties, and enables one to make three-way performance tradeoffs among privacy, convergence, and payment accuracy loss.

There are a few future directions. First, we assume that the (energy) cost of computation and communication is negligible. It is important to consider and analyze the impacts of such cost, and design economic mechanisms that is not only faithful but also elicits the right amount of efforts. Second, it is also interesting to design faithful algorithms and corresponding economic mechanisms for other more sophisticated federated learning architectures (e.g., multi-task federated learning [13]).

References

[1] M. Zhang, E. Wei, and R. Berry, "Faithful federated learning," in *Proc. 16th Workshop Econ. Netw., Syst. Comput. (NetEcon)*, 2021.

[2] R. Kelly, "Internet of Things data to top 1.6 zettabytes by 2020," Apr. 2015. [Online]. Available: https://campustechnology.com/articles/2015/04/15/internet-of-things-data-to-top-1-6-zettabytes-by-2020.aspx

[3] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, et al. "Communication-efficient learning of deep networks from decentralized data," arXiv:1602.05629, 2016.

[4] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," [Online]. Available: https://arxiv.org/abs/1811.03604.

[5] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, 2019.

[6] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proc. ACM Workshop on Artificial Intelligence and Security (AISec)*, 2019.

[7] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable Private Learning with PATE," [Online]. Available: https://arxiv.org/abs/1802.08908.

[8] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, "Distributed learning without distress: Privacy-preserving empirical risk minimization," *Proc. Advances in Neural Info. Process. Sys. (NIPS)*, 2018.

[9] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Trans. Industrial Informatics*, vol. 16, no. 10, pp. 6532-6542, 2019.

[10] X. Cao, J. Jinyuan, and N. Z. Gong, "Provably Secure Federated Learning against Malicious Clients," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 35, no. 8, pp. 6885-6893, 2021.

[11] T. D. Nguyen, *et al*., "BAFFLE: Towards resolving federated learning's dilemma-thwarting backdoor and inference attacks," under review by ICLR, 2021.

[12] R. D. Nowak M. G. Rabbat, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 798–808, 2005.

[13] V. Smith, C. K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," In Advances in Neural Information Processing Systems, pp. 4424–4434. 2017.

[14] R. E. Shostak L. Lamport and M. C. Pease. "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.,* vol. 4, pp. 382–401. 1982.

[15] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," arXiv preprint arXiv:1912.13445, 2019.

[16] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. Nitin Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, "Advances and open problems in federated learning," [Online]. Available: arXiv: 1912.04977.

[17] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19-25, January 2020.

[18] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205-1221, June 2019.

[19] H. H. Yang, Z. Liu, T. Q. S. Quek and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317-333, Jan. 2020.

[20] K. Yang, T. Jiang, Y. Shi and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022-2035, March 2020.

[21] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," [Online]. Available: https://arxiv.org/abs/1804.08333.

[22] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans Wireless Commun.*, vol. 20, no. 1, pp. 269-283, Jan. 2021.

[23] Z. Yang, M. Chen, W. Saad, C. S. Hong and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," accepted to *IEEE Trans. Wireless Commun.*.

[24] T. Zeng, O. Semiari, M. Chen, W. Saad, and M. Bennis, "Federated learning on the road: Autonomous controller design for connected and autonomous vehicles," [Online]. Available: https://arxiv.org/abs/1804.08333.

[25] M. M. Amiria, D. Gündüzb, S. R. Kulkarni and H. Vincent Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," accepted to *IEEE Trans. Wireless Commun.*

[26] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, "A survey of incentive mechanism design for federated learning", *IEEE Transactions on Emerging Topics in Computing,* 2021.

[27] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[28] W. Y. B. Lim, Z. Xiong, C. Miao, D. Niyato, Q. Yang, C. Leung, and H. V. Poor, "Hierarchical incentive mechanism design for federated machine learning in mobile networks," *IEEE Internet Things J.*, 2020.

[29] R. Zeng, S. Zhang, J. Wang, and X. Chu, "Fmore: An incentive scheme of multi-dimensional auction for federated learning in mec," in *Proc. IEEE ICDCS*, 2020.

[30] R. H. L. Sim, Y. Zhang, M. C. Chan, and B. K. H. Low, "Collaborative machine learning with incentive-aware model rewards," in *Proc. International Conference on Machine Learning (ICML)*, 2020.

[31] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications." *ACM Transaction Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[32] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[33] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "A crowdsourcing framework for on-device federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3241-3256, May 2020.

[34] 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.*, May 14, 2020.

[35] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[36] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A Stackelberg game perspective," *IEEE Netw. Lett.*, vol. 2, no. 1, pp. 23–27, Mar. 2020.

[37] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A sustainable incentive scheme for federated learning," *IEEE Intelligent Systems*, 2020.

[38] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A fairness-aware incentive scheme for federated learning," in *Proc. AAAI/ACM AIES,* pp. 393–399, 2020.

[39] N. Ding, Z. Fang, and J. Huang, "Optimal contract design for efficient federated learning with multi-dimensional private information," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 186-200, 2020.

[40] P. Sun, H. Che, Z. Wang, Y. Wang, T. Wang, L. Wu, and H. Shao. "Pain-FL: Personalized privacy-preserving incentive for federated learning," to appear in *IEEE J. Sel. Areas Commun.*, 2021.

[41] V. V. Vazirani, N. Nisanm, T. Roughgarden, and E. Tardos, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[42] D. C. Parkes, and J. Shneidman, "Distributed implementations of Vickrey-Clarke-Groves mechanisms," 2004.

[43] J. Feigenbaum, R. Sami, and S. Shenker, "Mechanism design for policy routing," *Distributed Computing*, vol. 18, no. 4, pp. 293-305, 2006.

[44] A. Petcu, B. Faltings, and D. C. Parkes, "M-DPOP: Faithful distributed implementation of efficient social choice problems." *J. Artif. Intell. Res. (JAIR)*, vol. 32, pp. 705-755, 2008.

[45] T. Tanaka, F. Farokhi and C. Langbort, "Faithful implementations of distributed algorithms and control laws," *IEEE Trans. Control Netw. Sys.*, vol. 4, no. 2, pp. 191-201, June 2017.

[46] V. Vapnik, "Principles of risk minimization for learning theory," in *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*, pp. 831-838, 1992.

[47] K. Sridharan, S. Shalev-Shwartz, and N. Srebro, "Fast rates for regularized objectives," in *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*, 21, pp.1545-1552, 2008.

[48] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *Proc. International Conference on Machine Learning (ICML)*, 2019.

[49] P. Milgrom and I. Segal, "Envelope theorems for arbitrary choice sets," *Econometrica*, vol. 70, no. 2, pp. 583-601, 2002.

[50] H. Karimi, J. Nutini, and M. W. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition," in: CoRRabs/1608.04636 (2016).

[51] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, vol. 3876, March 2006, pp. 265–284.

[52] S. Song, K. Chaudhuri and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proc. IEEE Global Conference on Signal and Information Processing*, Austin, USA, pp. 245-248, 2013.

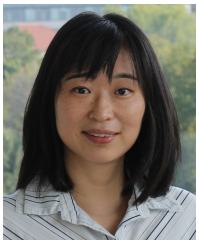
[53] M. J., Wainwright, *High-dimensional statistics: A non-asymptotic viewpoint*, vol. 48,. Cambridge University Press, 2019.

[54] M. Bun and T. Steinke. "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. Theory of Cryptography Conference*, 2016.

[55] 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.

[56] M. Zhang, E. Wei, and R. Berry, "Faithful edge federated learning: Scalability and Privacy", Technical Report. [Online]. Available: https://arxiv.org/abs/1804.08333.

Meng Zhang (S'15 – M'19) is an Assistant Professor with the Zhejiang University/University of Illinois at Urbana-Champaign Institute (ZJU-UIUC Institute), Zhejiang University. He has been a Postdoctoral Fellow with the Department of Electrical and Computer Engineering at Northwestern University from 2020 to 2021. He received his Ph.D. degree in Information Engineering from the Chinese University of Hong Kong in 2019. He was a visiting student research collaborator with the Department of Electrical Engineering at Princeton University from 2018 to 2019. His primary research interests include network economics and wireless networks, with a current emphasis on mechanism design and optimization for age of information and federated learning.



Ermin Wei is currently an Assistant Professor at the Electrical and Computer Engineering Department and Industrial Engineering and Management Sciences Department of Northwestern University. She completed her PhD studies in Electrical Engineering and Computer Science at MIT in 2014, advised by Professor Asu Ozdaglar, where she also obtained her M.S.. She received her undergraduate triple degree in Computer Engineering, Finance and Mathematics with a minor in German, from University of Maryland, College Park. Wei has received many awards, including the Graduate Women of Excellence Award, second place prize in Ernst A. Guillemen Thesis Award and Alpha Lambda Delta National Academic Honor Society Betty Jo Budson Fellowship. Her team also won the 2nd place in the Grid Optimization (GO) competition 2019, an electricity grid optimization competition organized by Department of Energy. Wei's research interests include distributed optimization methods, convex optimization and analysis, smart grid, communication systems and energy networks and market economic analysis.

Randall Berry (F'14) is the John A. Dever Professor and Chair of Electrical and Computer Engineering at Northwestern University. He is also a Principle Engineer with Roberson and Associates and has been on the technical staff of MIT Lincoln Laboratory. He received the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology in 1996 and 2000, respectively, and the BS degree from the University of Missouri Rolla in 1993. Dr. Berry is the recipient of a NSF CAREER award and an IEEE Fellow. He has served as an Editor for the IEEE Transactions on Wireless Communications from 2006 to 2009, and an Associate Editor for the IEEE Transactions on Information Theory from 2009 to 2011. He is currently a Division Editor for the Journal of Communications and Networks and an Area editor for the IEEE Open Journal of the Communications Society. He has also been a guest editor for special issues of the IEEE Journal on Selected Topics in Signal Processing, the IEEE Transactions on Information Theory and the IEEE Journal on Selected Areas in Communications. He has served on the program and organizing committees of numerous conferences including serving as a chair of the 2012 IEEE Communication Theory Workshop, a TPC chair of 2010 IEEE ICC Wireless Networking Symposium, and a TPC chair of the 2018 ACM Mobihoc conference.