A Blockchain-Empowered Incentive Mechanism for Cross-Silo Federated Learning

Ming Tang, *Member, IEEE*, Fu Peng, and Vincent W.S. Wong, *Fellow, IEEE*

Abstract—In cross-silo federated learning (FL), organizations cooperatively train a global model with their local datasets. However, some organizations may act as free riders such that they only contribute a small amount of resources but can obtain a high-accuracy global model. Meanwhile, some organizations can be business competitors, and they do not trust each other or any third-party entity. In this work, our goal is to design a framework that motivates efficient cooperation among organizations without the coordination of a central entity. To this end, we propose a blockchain-empowered incentive mechanism framework for cross-silo FL. Under this incentive mechanism framework, we develop a distributed algorithm that enables organizations to achieve social efficiency, individual rationality, and budget balance without private information of the organizations. Our proposed algorithm has a proven convergence guarantee and empirically achieves a higher convergence rate than a benchmark method. Moreover, we propose a transaction minimization algorithm to reduce the number of transactions made among organizations in the blockchain. This algorithm is proven to achieve a performance no worse than twice the minimum value. The experimental results in a testbed show that our proposed framework enables organizations to achieve social efficiency within a relatively short iterative process.

Index Terms—Federated learning, blockchain, incentive mechanism, game theory, resource allocation.

1 Introduction

1.1 Background and Motivation

Cross-silo federated learning (FL) [2] enables a set of organizations (e.g., network operators, institutions) to cooperatively train a global model using their local datasets. During the training phase, each organization periodically downloads the global model from a certain entity (e.g., a central server, a blockchain network [3]), updates its local model by training the downloaded global model with its local dataset, and uploads the model updates to the entity for global model updating. Since each organization does not need to share its local dataset with other entities, data privacy can be preserved. Cross-silo FL can support the cooperation among network operators for applications such as mobile traffic prediction [4] and content popularity prediction [5]. Meanwhile, there have been various industrial applications, such as Owkin [6] for medical data, MELLODDY [7] for drug discovery, and Ichnite [8] for general dataset analysis. Moreover, algorithms have been developed to enable cross-silo FL. McMahan *et al.* [9] proposed federated average (FedAvg) algorithm. Some works proposed algorithms to improve the convergence rate [10]–[12] and system structure [13]. Comprehensive surveys can be found in [2], [14], [15].

In contrast to the cross-device FL where a central entity (e.g., an organization) recruits devices for local training, cross-silo FL involves cooperation among multiple organizations and has four distinctive features. First, each organization usually has a large number of data samples. Thus, the training speed and cost depend on the processing capacity (e.g., number of computing servers, processing frequency of computing servers) that the organizations use for training. Second, in cross-silo FL, organizations perform local training to obtain a high-accuracy global model so that the revenue of their services can be improved. Thus, there may not exist any central entity (as in cross-device FL) that can compensate for the training cost of the organizations and motivate their cooperation. Third, due to the non-excludable nature of the global model (i.e., no organization can prevent another organization from receiving the global model during the FL process), an organization can become a free rider. That is, an organization that allocates a small amount of processing capacity for local training may be able to obtain a high-accuracy global model, due to the high processing capacity provided by other organizations. Fourth, some organizations can be business competitors. They may not trust each other or any third-party entity. These four features make it necessary to design an incentive mechanism that can motivate efficient cooperation among the organizations, and the operation of the incentive mechanism should not rely on the assistance of any central coordinator.

Blockchain together with smart contract [16] can perform secure executions without any central coordinator. In this work, we aim to propose a blockchain-empowered incentive

- *Ming Tang is with the Department of Computer Science and Engineering and the Research Institute of Trustworthy Autonomous Systems at Southern University of Science and Technology, Shenzhen, China. E-mail: tangm3@sustech.edu.cn*
- *Fu Peng is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. Email: 12231135@mail.sustech.edu.cn*
- *Vincent W.S. Wong is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada. E-mail: vincentw@ece.ubc.ca*

Manuscript received on Apr. 25, 2023; revised on Nov. 30, 2023; accepted on Jan. 23, 2024. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, Digital Research Alliance of Canada (alliancecan.ca), Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515012819, and the National Natural Science Foundation of China under Grant 62202214. (Corresponding author: Ming Tang)
Part of this work has been published in Proc. of IEEE INFOCOM [1].

mechanism framework that motivates efficient cooperation of the organizations in terms of their choices of processing capacities in cross-silo FL. Note that those existing works which applied blockchain techniques to the FL training process [3], [17]–[20] are not directly applicable to our work, as they did not investigate the incentive mechanism design and how the mechanism can be incorporated in the blockchain. Designing such a blockchain-empowered incentive mechanism framework has several challenges.

Challenge 1. *The incentive mechanism framework needs to address the free-rider attack resulting from the public goods nature of the global model in cross-silo FL.*

Specifically, public goods are defined as those goods that are non-excludable and non-rivalrous [21]. In cross-silo FL, the global model is non-excludable, because during the FL training process, the global model is sent to all organizations periodically for local training. Meanwhile, the global model is non-rivalrous, i.e., there is no competition among the organizations for owning a copy of the global model. The public goods nature of the global model leads to the free-rider attack in cross-silo FL, i.e., some organizations may perform local training with a low processing capacity that does not optimize the social welfare. This is because in practical systems, some organizations may have lower willingness to improve the accuracy of the global model than some other organizations, since they can earn only little revenue through such a global model improvement (e.g., due to their small customer population). Note that the free-rider attack may result from the misbehavior of organizations during two processes: (i) when the organizations negotiate about their processing capacities and compensations, and (ii) when the organizations perform FL. Although there are various incentive mechanisms for cross-device FL, including pricing-based schemes [22], [23], auction-based mechanisms [24]–[27], and contract-based mechanisms [28]–[30], those mechanisms are not applicable to cross-silo FL due to the public goods nature. On the other hand, there are existing works on public goods in other application scenarios, such as multicasting [31] and energy harvesting [32] in wireless networks. However, those mechanisms cannot be directly applied in cross-silo FL, because the incentive mechanism design in cross-silo FL involves a nonconvex problem formulation, which poses additional difficulties.

Challenge 2. *The operation of the blockchain-empowered framework should (i) not rely on the private information (e.g., valuation, cost) of the organizations and (ii) be lightweight.*

In cross-silo FL, the private information of the organizations is usually unknown to the public. Thus, the incentive mechanism should not make use of such information. Although auction (e.g., [33]) is a commonly used approach to address the private information issue, it is not applicable in cross-silo FL due to the public goods feature. In addition, there is a fee incurred in each execution step in smart contract (e.g., gas fee in Ethereum [34]). The fee depends on the computational resources required for performing the execution step. Thus, the operation of the incentive mechanism performed by smart contract has to be lightweight such that it has a low computational complexity and consumes a small amount of computational resources. Although pricing-based [22], [23], auction-based [24]–[27], and contract-based mechanisms [28]–[30] can achieve lightweight execution in smart contract, they are not applicable to cross-silo FL due to Challenge 1.

Challenge 3. *The blockchain-empowered framework should minimize the monetary transfer between organizations.*

Different from cross-device FL where the central entity recruits mobile devices for training, in cross-silo FL, organizations need to pay each other to compensate for their training cost. Since the blockchain charges a fee for each transaction, the organizations need to minimize the number of transactions in order to reduce the transaction cost.

1.2 Related Works

Some recent works proposed incentive mechanisms for cross-silo FL. Wu *et al.* in [35] analyzed the market share of the organizations participating in FL. However, they did not consider the training cost of the organizations. Zhang *et al.* in [36] proposed a mechanism that motivates organizations to truthfully submit their model updates. Zhang *et al.* in [37] considered a repeated game setting and proposed a scheme where the organizations deviating from certain strategies will be punished. However, the works in [36], [37] did not consider the processing capacities that the organizations should allocate for local training. Huang *et al.* in [38] proposed a minimum threshold-based incentive mechanism to prevent the free-rider attack, where the organizations contributing resources lower than a threshold will be removed from the system. Lim *et al.* in [39] proposed a coalitional game-based mechanism to motivate the organizations to form FL coalitions. The algorithms proposed in [38], [39] rely on the knowledge of the payoff of the organizations, which did not address Challenge 2 (i). Furthermore, those approaches in [35]–[39] rely on the assistance of a trusted central coordinator. It may be difficult to incorporate those approaches in a blockchain-empowered system due to the lightweight requirement in Challenge 2 (ii) and the minimization of monetary transfer requirement in Challenge 3.

Some existing works (e.g., [40], [41]) considered secure aggregation of the global model, while they did not consider incentivizing efficient cooperation. Other recent works (e.g., [23], [42]–[45]) have considered blockchain-empowered incentive mechanisms for FL. However, in those works, there exists a task requester who requests the global model and provides payment to the organizations (in cross-silo FL) or devices (in cross-device FL). Thus, those mechanisms do not address the free-rider attack (Challenge 1) in cross-silo FL.

1.3 Solution and Contributions

In this work, we propose a blockchain-empowered incentive mechanism framework for cross-silo FL that addresses Challenges 1–3. Our proposed mechanism helps the organizations to address the following questions: (i) How much processing capacity should each organization allocate for local training? (ii) How much should each organization be compensated by other organizations for its local training? Our main contributions are summarized as follows:

- **Incentive Mechanism Design Problem Formulation:** We formulate a social welfare maximization

problem for cross-silo FL, which is a nonconvex problem. Then, we formulate an incentive mechanism design problem, considering the public goods.

- **An Incentive Mechanism for Cross-Silo FL:** We propose an incentive mechanism that addresses the free-rider attack (Challenge 1). Given our proposed mechanism, the interaction between the organizations is formulated as a non-cooperative game. We prove that the Nash equilibrium (NE) of the game has several properties, including *social efficiency* (i.e., the social welfare is maximized), *individual rationality* (i.e., each organization is no worse off by participating in FL), and *budget balance* (i.e., no third party investment is required). Moreover, we prove that after reaching an agreement based on our proposed mechanism, the organizations prefer to behave according to their agreement during the FL process.
- **A Blockchain-Empowered Framework:** We propose a blockchain-empowered framework for deploying our proposed incentive mechanism. Under this framework, we develop a distributed algorithm that enables organizations to achieve social efficiency, individual rationality, and budget balance. This algorithm does not rely on the private information of the organizations (Challenge 2 (i)), and its execution steps in the smart contract only involve simple operations (e.g., replacement, addition, comparison) and have polynomial complexity (Challenge 2 (ii)). In addition, we propose a transaction minimization algorithm that reduces the number of monetary transfer among organizations in the blockchain (Challenge 3).
- **Performance Evaluation:** The simulation results verify that our proposed distributed algorithm can achieve social efficiency, and it converges faster than the conventional Lagrangian method [46, Section 8.1]. Moreover, our proposed transaction minimization algorithm achieves comparable performance as the optimal solution but has a significantly lower computational time. Furthermore, we build a demonstration system of our blockchain-empowered incentive mechanism framework with Ethereum [34]. The experimental results show that it takes tens of minutes for our proposed algorithm to converge to the NE in real-world blockchain systems, where this duration is relatively short when compared with the duration of the FL process (e.g., 1-10 days [2]).

This paper is organized as follows. We present the system model in Section 2 and the incentive mechanism in Section 3. The blockchain-empowered framework is given in Section 4. We conduct simulations and experiments in Section 5. Section 6 concludes this work. Notations: We use $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{Z}_+$ to denote the sets of real numbers, nonnegative real numbers, and nonnegative integers, respectively.

2 System Model

An overview of our considered cross-silo FL system is shown in Fig. 1. Specifically, organizations first determine their processing capacities and unit monetary transfer per training round using an incentive mechanism. Then, they train a global model using FL. After the FL process, organizations need to make transactions between each other to compensate for their training cost. In this system, the free-rider attack corresponds to the attack where an organization performs local training with the processing capacity that does not optimize the social welfare. Such an attack may result from the misbehavior of the organizations during either the incentive mechanism or FL process. Furthermore, in this work, we introduce a blockchain network that acts as a decentralized agent for coordinating the incentive mechanism, FL, and transaction processes. It is responsible for enabling message and transaction exchange among the organizations, performing simple executions required by the aforementioned processes, and sending control messages. Existing works investigated how the blockchain network coordinates the FL process [3], [17]–[20]. We will describe how the blockchain network can coordinate the incentive mechanism and transaction processes in Section 4.

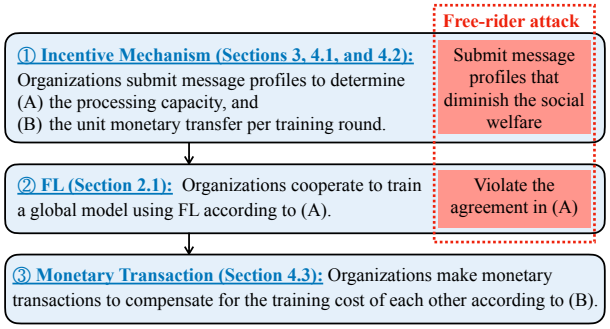


Fig. 1. Organizations perform incentive mechanism, FL, and transaction processes. The free-rider attack may occur due to the misbehavior of the organizations during either the incentive mechanism or FL process.

In this section, we present the FL process and payoff of the organizations. Then, we formulate the social welfare maximization and incentive mechanism design problems.

2.1 FL Process

We consider a scenario with N organizations, denoted by set $\mathcal{N} = \{0, \ldots, N-1\}$. Each organization has its own local dataset. Let $\mathcal{S}_n$ denote the local dataset of organization $n \in \mathcal{N}$. Let S_n denote the number of data units in set $\mathcal{S}_n$. The organizations cooperate to train a global model with their local datasets using a synchronous FL algorithm, such as [9]–[11]. Let $\boldsymbol{\omega}$ denote the global model. The organizations aim to find the optimal weights of the global model $\boldsymbol{\omega}^*$ that minimize the expected loss $L(\boldsymbol{\omega})$ over the datasets [9], [11]:

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} \left\{ L(\boldsymbol{\omega}) \triangleq \sum_{n\in\mathcal{N}} \frac{S_n}{\sum_{n'\in\mathcal{N}} S_{n'}} l(\boldsymbol{\omega}; \mathcal{S}_n) \right\}, \quad (1)$$

where $l(\boldsymbol{\omega}; \mathcal{S}_n)$ is the loss over dataset $\mathcal{S}_n$ given $\boldsymbol{\omega}$.

At the beginning of the FL algorithm, the coordinator[1] first randomly initializes the global model $\boldsymbol{\omega}^0$. The FL algorithm iterates for multiple training rounds. In training round r, each organization $n \in \mathcal{N}$ downloads the global model $\boldsymbol{\omega}^{r-1}$ (determined in training round $r-1$) from

1. During FL process, the coordinator can be either a blockchain network or a central server. The choice of such a coordinator does not affect the properties of our incentive mechanism in Section 3.

the coordinator. It then performs K local updates over the downloaded global model with its dataset $\mathcal{S}_n$, e.g., the local update can correspond to a mini-batch stochastic gradient descent [9]. The updated model is the local model of organization n in training round r, where $\boldsymbol{\omega}_n^r$ denotes the local model. After that, organization n uploads $\boldsymbol{\omega}_n^r$ to the coordinator. The coordinator updates the global model based on the received local models, e.g., by taking a weighted average over the received local models from the organizations $\boldsymbol{\omega}^r = \sum_{n\in\mathcal{N}} S_n\boldsymbol{\omega}_n^r/(\sum_{n'\in\mathcal{N}} S_{n'})$ [9].

Let D_n denote the number of floating point operations (FLOPs) required by organization $n \in \mathcal{N}$ to process one data unit. Let f_n (in FLOPs per second) denote the processing capacity used by organization n for its local training. Let vector $\boldsymbol{f} = (f_n, n \in \mathcal{N})$. Note that we adopt a standard processing capacity model, which is applicable to model both CPU and GPU processors [47]. Let T_n^{UL} and T_n^{DL} (in seconds) denote the time that organization n is required for uploading and downloading the model updates in each training round, respectively. Since we consider a synchronous FL algorithm where the coordinator updates the global model only after it has received all the local models in a training round, the duration of each training round is given by

$$\tau(\boldsymbol{f}) = \max_{n\in\mathcal{N}} \left\{ \frac{S_n D_n K}{f_n} + T_n^{\text{UL}} + T_n^{\text{DL}} \right\}, \tag{2}$$

which is the maximum processing and transmission duration in each training round among all organizations.

We consider that there is a fixed total training time $T \in \mathbb{R}_+$, where T is usually much larger than the uplink and downlink duration (i.e., $T_n^{\text{UL}} + T_n^{\text{DL}}$) to ensure multiple rounds of training. This corresponds to the scenario where the organizations have a deadline for the FL process. The number of training rounds is equal to the total training time divided by the duration of each training round:

$$r(\boldsymbol{f}) = T/\tau(\boldsymbol{f}). \tag{3}$$

Similar to [22], [44], we do not round up $r(\boldsymbol{f})$. This is reasonable since $r(\boldsymbol{f})$ is usually large in practice, so the difference between $r(\boldsymbol{f})$ and its rounded value is negligible.

2.2 Payoff of the Organizations

2.2.1 Utility

As in the existing work [22], we define the utility of each organization as a function of the precision of the trained global model. The precision of the trained global model is defined as the difference between the expected loss of the trained global model (after $r(\boldsymbol{f})$ training rounds) and the minimum expected loss [10], [11], [22], i.e., $L(\boldsymbol{\omega}^{r(\boldsymbol{f})}) - L(\boldsymbol{\omega}^*)$. Note that a smaller precision implies a smaller loss of the trained global model and hence a better fit of the model to the datasets. Let $\epsilon(r(\boldsymbol{f}))$ denote the precision of the trained global model with $r(\boldsymbol{f})$ training rounds.[2] We consider a general formulation of precision function $\epsilon(r(\boldsymbol{f}))$ that satisfies the following assumption:

2. Since we focus on the organizations' choices of processing capacities (which affect the number of training rounds $r(\boldsymbol{f})$), the precision function $\epsilon(r(\boldsymbol{f}))$ takes $r(\boldsymbol{f})$ as an argument. Although the precision also depends on other factors such as the number of data samples used for training, we omit those factors in our formulation for simplicity.

Assumption 1. *Function $\epsilon(r(\boldsymbol{f}))$ is a continuously differentiable function that is non-increasing and convex in $r(\boldsymbol{f})$. In addition, $\epsilon(0)$ has a finite value.*

Intuitively, as $r(\boldsymbol{f})$ increases, the degree that the global model fits the datasets is non-decreasing, and the marginal decrease of the precision reduces. Note that this assumption does not limit to the setting of independent and identically distributed (IID) datasets in FL. For example, based on [11], with a strongly convex loss function $L(\boldsymbol{\omega})$, $\epsilon(r(\boldsymbol{f}))$ under non-IID datasets can be modeled as follows:

$$\epsilon(r(\boldsymbol{f})) = \frac{\epsilon_0}{\epsilon_1 + K r(\boldsymbol{f})}, \tag{4}$$

where the positive coefficients ϵ_0 and ϵ_1 can be determined based on the loss function, the neural network structure, and the distribution of the datasets [11]. It is easy to verify that $\epsilon(r(\boldsymbol{f}))$ in (4) satisfies Assumption 1. We use equation (4) for the performance evaluation in Section 5.

The utility of organization $n \in \mathcal{N}$ is its valuation on the difference between the precision of the global model without training (i.e., $\epsilon(0)$) and that after training:

$$U_n(r(\boldsymbol{f})) = u_n \left(\epsilon(0) - \epsilon(r(\boldsymbol{f}))\right),\ n \in \mathcal{N}, \tag{5}$$

where u_n (in dollars per unit of loss) is the unit revenue that organization n can earn from its market by using the trained global model. The value of u_n of organization n may be unknown to the coordinator and other organizations.

2.2.2 Cost and Payoff

The cost of an organization is defined as follows:

$$\begin{aligned} C_n(f_n, r(\boldsymbol{f})) = {} & (C_n^{\text{UL}} + C_n^{\text{DL}})\, r(\boldsymbol{f}) + C_n^{\text{invt}} f_n \\ & + C_n^{\text{comp}} (f_n)^2 S_n D_n K r(\boldsymbol{f}),\ n \in \mathcal{N}, \end{aligned} \tag{6}$$

where the parameters C_n^{UL} and C_n^{DL} are the operating costs for uploading and downloading the model updates in each training round, respectively. The product $C_n^{\text{invt}} f_n$ corresponds to the investment cost (e.g., leasing servers) per processing capacity, where the linearity is due to the linear server leasing rate [48]. The term $C_n^{\text{comp}}(f_n)^2 S_n D_n K$ is the operating cost of organization n for performing local training in one training round, where the quadratic form is due to the widely adopted quadratic energy consumption model for both CPU and GPU processors [47]. The cost function $C_n(f_n, r(\boldsymbol{f}))$ of organization n may not be known by the coordinator and other organizations.

Let m_n (in dollars) denote the monetary transfer to organization $n \in \mathcal{N}$. If m_n is positive, then organization n receives m_n from other organizations. If m_n is negative, then organization n pays $|m_n|$ to some other organizations. The payoff of organization $n \in \mathcal{N}$ is defined as follows:

$$V_n(f_n, r(\boldsymbol{f}), m_n) = U_n(r(\boldsymbol{f})) - C_n(f_n, r(\boldsymbol{f})) + m_n. \tag{7}$$

2.3 Problem Formulation

From the social welfare perspective, the organizations should choose the processing capacity vector $\boldsymbol{f}$ that maximizes the social welfare of the system:

$$\underset{\boldsymbol{f}}{\text{maximize}} \quad \sum_{n\in\mathcal{N}} \left(U_n(r(\boldsymbol{f})) - C_n(f_n, r(\boldsymbol{f}))\right) \tag{8a}$$

$$\text{subject to} \quad f_n \in [0, f_n^{\max}],\ n \in \mathcal{N}, \tag{8b}$$

where $f_n^{\max}$ denotes the maximum processing capacity of organization n. It is challenging to maximize the social welfare for two reasons. First, problem (8) is nonconvex, since $C_n(f_n, r(\boldsymbol{f}))$ is nonconvex in $\boldsymbol{f}$. Second, each organization may only be interested in maximizing its own payoff. Let $\boldsymbol{f}^* \triangleq (f_n^*, n \in \mathcal{N})$ denote the optimal solution to problem (8). The free-rider attack is defined as follows:

Definition 1 (Free-Rider Attack)**.** *The free-rider attack is an attack where there exists an organization $n' \in \mathcal{N}$ that performs local training with a processing capacity $f_{n'} \neq f_{n'}^*$.*

We aim to propose a blockchain-empowered incentive mechanism framework to prevent the free-rider attack and achieve social efficiency. As shown in Fig. 1, the organizations first determine their processing capacities and monetary transfer using an incentive mechanism in ①:

- Each organization $n \in \mathcal{N}$ announces a message profile (γ_n, π_n) to other organizations. Message γ_n indicates the number of training rounds that organization n is willing to have. Message π_n indicates the unit monetary transfer per training round that organization n is willing to receive or pay. Let vectors $\boldsymbol{\gamma} \triangleq (\gamma_n, n \in \mathcal{N})$ and $\boldsymbol{\pi} \triangleq (\pi_n, n \in \mathcal{N})$.
- Based on the message profiles $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$, organizations determine their processing capacity $\boldsymbol{f}(\boldsymbol{\gamma}) \triangleq (f_n(\boldsymbol{\gamma}), n \in \mathcal{N})$ and unit monetary transfer per training round $\boldsymbol{\zeta}(\boldsymbol{\pi}) \triangleq (\zeta_n(\boldsymbol{\pi}), n \in \mathcal{N})$.[3]

Then, in ②, organizations are expected to cooperate on FL with the determined processing capacity $\boldsymbol{f}(\boldsymbol{\gamma})$. However, an organization may use a processing capacity that differs from the determined one. Suppose organizations perform local training with processing capacity $\boldsymbol{f}$. In ③, the monetary transfer to organization n is equal to the actual number of training rounds $r(\boldsymbol{f})$ multiplied by the determined unit monetary transfer to organization n, i.e., $m_n = \zeta_n(\boldsymbol{\pi}) r(\boldsymbol{f})$.

Let Γ and Π denote the space of $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$, respectively. To motivate efficient cooperation through incentive mechanism design, we need to specify the mappings $f_n : \Gamma \to \mathbb{R}^+$ and $\zeta_n : \Pi \to \mathbb{R}^+$ for $n \in \mathcal{N}$ such that (i) organizations make an agreement on performing local training using the optimal processing capacity $\boldsymbol{f}^*$ in the incentive mechanism process, and (ii) they follow the agreement during the FL process. We define the properties formally as follows.

2.3.1 Properties Required in Incentive Mechanism Process

During the incentive mechanism process in ①, organizations decide on the message profiles to announce. The strategic interaction among the organizations can be modeled as a non-cooperative game. Let $(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$ denote an NE of the game. That is, given the NE message profiles of other organizations, organization $n \in \mathcal{N}$ cannot increase its payoff by announcing a message profile that is different from its NE message profile $(\gamma_n^{\mathrm{NE}}, \pi_n^{\mathrm{NE}})$. Note that such an NE is defined based on the setting that all organizations will follow the determined processing capacity during the FL process. This setting always holds, because it is considered as a required property during the FL process (see Section 2.3.2).

3. The determined unit monetary transfer $\zeta_n(\boldsymbol{\pi})$ may be different from the value of π_n specified by organization n in its message profile.

We define $m_n(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}}) \triangleq \zeta_n(\boldsymbol{\pi}^{\mathrm{NE}}) r(\boldsymbol{f}(\boldsymbol{\gamma}^{\mathrm{NE}}))$, i.e., the monetary transfer to organization n if everyone follows the determined processing capacity $\boldsymbol{f}(\boldsymbol{\gamma}^{\mathrm{NE}})$. Our incentive mechanism needs to satisfy the following properties:

P1 Social efficiency: The processing capacity under NE is the optimal solution of problem (8), i.e., $\boldsymbol{f}(\boldsymbol{\gamma}^{\mathrm{NE}}) = \boldsymbol{f}^*$.

P2 Individual rationality: Each organization is no worse off through participating in cross-silo FL, i.e., $V_n(f_n(\boldsymbol{\gamma}^{\mathrm{NE}}), r(\boldsymbol{f}(\boldsymbol{\gamma}^{\mathrm{NE}})), m_n(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})) \geq 0$ for $n \in \mathcal{N}$.

P3 Budget balance: The monetary transfer can operate among the organizations without any third-party investment. That is, $\sum_{n \in \mathcal{N}} m_n(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}}) = 0$.

2.3.2 Property Required in FL Process

We use $f_n^{\mathrm{NE}} \triangleq f_n(\boldsymbol{\gamma}^{\mathrm{NE}})$ and $\zeta_n^{\mathrm{NE}} \triangleq \zeta_n(\boldsymbol{\pi}^{\mathrm{NE}})$ to denote the processing capacity and unit monetary transfer per training round of organization $n \in \mathcal{N}$ under NE message profiles, respectively. Let vector $\boldsymbol{f}^{\mathrm{NE}} \triangleq (f_n^{\mathrm{NE}}, n \in \mathcal{N})$. Let vector $\boldsymbol{f}_{-n}^{\mathrm{NE}} \triangleq (f_{n'}^{\mathrm{NE}}, n' \in \mathcal{N} \setminus \{n\})$ denote the processing capacity of all organizations excluding organization n's. We use the notations $(f_n^{\mathrm{NE}}, \boldsymbol{f}_{-n}^{\mathrm{NE}})$ and $\boldsymbol{f}^{\mathrm{NE}}$ interchangeably. During the FL process in ②, organizations perform FL using the determined processing capacity $\boldsymbol{f}^{\mathrm{NE}}$. Thus, our incentive mechanism needs to satisfy the following property:

P4 Fulfillment of agreement: Each organization weakly prefers to follow the determined processing capacity vector $\boldsymbol{f}^{\mathrm{NE}}$, i.e., $V_n(f_n^{\mathrm{NE}}, r(f_n^{\mathrm{NE}}, \boldsymbol{f}_{-n}^{\mathrm{NE}}), \zeta_n^{\mathrm{NE}} r(f_n^{\mathrm{NE}}, \boldsymbol{f}_{-n}^{\mathrm{NE}})) \geq V_n(f_n, r(f_n, \boldsymbol{f}_{-n}^{\mathrm{NE}}), \zeta_n^{\mathrm{NE}} r(f_n, \boldsymbol{f}_{-n}^{\mathrm{NE}}))$ for all $f_n \in \mathbb{R}_+$, $n \in \mathcal{N}$.

Properties P1 and P4 prevent the free-rider attack in the incentive mechanism and FL processes, respectively.

3 Incentive Mechanism for Cross-Silo FL

We design an incentive mechanism for cross-silo FL that satisfies properties P1−P4. Our proposed incentive mechanism is inspired by the existing mechanisms for public goods in other application scenarios, e.g., [31], [32]. Different from those existing mechanisms, our proposed mechanism can address the following new challenges. First, the associated social welfare maximization problem (8) is nonconvex. Second, each organization is both a producer who contributes resources for local training and a consumer who utilizes the trained global model. In the following, we first propose an incentive mechanism. We then analyze its properties.

3.1 Incentive Mechanism Design

The incentive mechanism for cross-silo FL is as follows.

Mechanism 1 (Incentive Mechanism for Cross-Silo FL)**.** *Each organization $n \in \mathcal{N}$ announces its message profile (γ_n, π_n):*

- *$\gamma_n \in [0, \bar{r}]$ indicates the number of training rounds that organization n is willing to have. Here, $\bar{r} \triangleq \max_{f_n \in [0, f_n^{\max}], n \in \mathcal{N}} r(\boldsymbol{f}) = T / \max_{n \in \mathcal{N}} \{S_n D_n K / f_n^{\max} + T_n^{\mathrm{UL}} + T_n^{\mathrm{DL}}\}$ denotes the maximum number of training rounds.*
- *$\pi_n \in \mathbb{R}$ indicates the unit monetary transfer per training round that organization n is willing to receive or pay. If π_n is positive, then organization n desires to receive π_n from other organizations. If π_n is negative, then organization n is willing to pay $|\pi_n|$ to some other organizations.*

Based on the announced $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$:

- *The number of training rounds that each organization needs to perform $\widetilde{r}(\boldsymbol{\gamma}) = \sum_{n\in\mathcal{N}} \gamma_n/N$.*
- *The processing capacity that organization n should use in its local training is given by*

$$f_n(\boldsymbol{\gamma}) = f_n^{\circ}(\widetilde{r}(\boldsymbol{\gamma})) \triangleq \frac{S_n D_n K}{T/\widetilde{r}(\boldsymbol{\gamma}) - T_n^{\mathrm{UL}} - T_n^{\mathrm{DL}}}, \quad (9)$$

  where $f_n^{\circ}(\widetilde{r}(\boldsymbol{\gamma}))$ is the processing capacity that organization n should use to achieve $\widetilde{r}(\boldsymbol{\gamma})$ training rounds.
- *The unit monetary transfer (per training round) to organization n is given by*

$$\zeta_n(\boldsymbol{\pi}) = \pi_{\mu(n+1)} - \pi_{\mu(n+2)}, \quad (10)$$

  where $\mu(n+1)$ is equal to $n+1$ modulo N. That is, the unit monetary transfer to organization n is equal to the difference between the unit monetary transfer announced by the organizations with indices $\mu(n+1)$ and $\mu(n+2)$.
- *After FL process, the monetary transfer to organization n is equal to the unit monetary transfer multiplied by the number of training rounds r during FL process, i.e., $\zeta_n(\boldsymbol{\pi})r$. Thus, if each organization performs local training using the processing capacity defined in* (9)*, then*

$$m_n(\boldsymbol{\gamma}, \boldsymbol{\pi}) \triangleq \zeta_n(\boldsymbol{\pi})\widetilde{r}(\boldsymbol{\gamma}). \quad (11)$$

Intuitively, in Mechanism 1, to address the public goods feature (i.e., non-excludable, non-rivalrous), each organization $n \in \mathcal{N}$ is treated equally regardless of the value of π_n. In Mechanism 1, the number of training rounds that each organization has to perform (i.e., $\widetilde{r}(\boldsymbol{\gamma})$) is equal to the average value of the number of training rounds that the organizations are willing to have. The processing capacity of each organization has to lead to $\widetilde{r}(\boldsymbol{\gamma})$ training rounds. The definition of monetary transfer in (11) has three features that make our proposed incentive mechanism satisfy properties P1−P4. First, the payoff of organization n does not rely on the choice of π_n. Second, the payoff of organization n is linear in the number of training rounds. Third, the summation of the monetary transfer to all organizations, i.e., the summation of $m_n(\boldsymbol{\gamma}, \boldsymbol{\pi})$ over all $n \in \mathcal{N}$, is always equal to zero. Moreover, we will show in Section 4.2.2 that equations (10) and (11) essentially motivate each organization n to choose its value of γ_n that leads to social efficiency, even when the organizations have private information.

3.2 Analysis of the Strategies of the Organizations

We first define the game of the organizations. Then, we derive an NE and the properties of Mechanism 1.

3.2.1 Game of the Organizations

Given Mechanism 1, each organization can determine its message profile to maximize its payoff. Such strategic interaction can be modeled as a non-cooperative game. Recall that this game is defined under the setting that organizations perform local training using the processing capacity in (9). This is reasonable because property P4 holds under our proposed mechanism, which will be proven in Section 3.2.3.

Game 1 (Message Profile Announcement)**.**

- *Player: all organizations $n \in \mathcal{N}$.*
- *Strategy: message profile (γ_n, π_n) with $\gamma_n \in [0, \bar{r}]$ and $\pi_n \in \mathbb{R}$ for each organization $n \in \mathcal{N}$.*
- *Payoff function: $V_n(f_n(\boldsymbol{\gamma}), \widetilde{r}(\boldsymbol{\gamma}), m_n(\boldsymbol{\gamma}, \boldsymbol{\pi}))$ for $n \in \mathcal{N}$. Note that $V_n(f_n(\boldsymbol{\gamma}), r(\boldsymbol{f}(\boldsymbol{\gamma})), m_n(\boldsymbol{\gamma}, \boldsymbol{\pi}))$ is equal to $V_n(f_n(\boldsymbol{\gamma}), \widetilde{r}(\boldsymbol{\gamma}), m_n(\boldsymbol{\gamma}, \boldsymbol{\pi}))$ according to* (9)*.*

Let $(\boldsymbol{\gamma}_{-n}, \boldsymbol{\pi}_{-n})$ denote the message profiles announced by all organizations excluding organization $n \in \mathcal{N}$, i.e., $\boldsymbol{\gamma}_{-n} \triangleq (\gamma_{n'}, n' \in \mathcal{N} \setminus \{n\})$ and $\boldsymbol{\pi}_{-n} \triangleq (\pi_{n'}, n' \in \mathcal{N} \setminus \{n\})$. For simplicity, we will use $\boldsymbol{f}(\gamma_n, \boldsymbol{\gamma}_{-n})$ and $\boldsymbol{f}(\boldsymbol{\gamma})$, $\widetilde{r}(\gamma_n, \boldsymbol{\gamma}_{-n})$ and $\widetilde{r}(\boldsymbol{\gamma})$, as well as $m_n(\gamma_n, \pi_n, \boldsymbol{\gamma}_{-n}, \boldsymbol{\pi}_{-n})$ and $m_n(\boldsymbol{\gamma}, \boldsymbol{\pi})$ interchangeably. The NE of Game 1 is defined as follows.

Definition 2 (Nash equilibrium)**.** *An NE of Game 1 is a message profile $(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$ that satisfies*

$$\begin{aligned} &V_n(f_n(\boldsymbol{\gamma}^{\mathrm{NE}}), \widetilde{r}(\boldsymbol{\gamma}^{\mathrm{NE}}), m_n(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})) \\ &\geq V_n(f_n(\gamma_n, \boldsymbol{\gamma}_{-n}^{\mathrm{NE}}), \widetilde{r}(\gamma_n, \boldsymbol{\gamma}_{-n}^{\mathrm{NE}}), m_n(\gamma_n, \pi_n, \boldsymbol{\gamma}_{-n}^{\mathrm{NE}}, \boldsymbol{\pi}_{-n}^{\mathrm{NE}})), \\ &\qquad \gamma_n \in [0, \bar{r}], \pi_n \in \mathbb{R}, n \in \mathcal{N}. \end{aligned} \quad (12)$$

Based on (10) and (11), we have $m_n(\gamma_n, \pi_n, \boldsymbol{\gamma}_{-n}^{\mathrm{NE}}, \boldsymbol{\pi}_{-n}^{\mathrm{NE}}) = m_n(\gamma_n, \pi_n^{\mathrm{NE}}, \boldsymbol{\gamma}_{-n}^{\mathrm{NE}}, \boldsymbol{\pi}_{-n}^{\mathrm{NE}})$ for $\pi_n \in \mathbb{R}$ for any $N \geq 3$. Hence, inequality (12) is equivalent to

$$\begin{aligned} &V_n(f_n(\boldsymbol{\gamma}^{\mathrm{NE}}), \widetilde{r}(\boldsymbol{\gamma}^{\mathrm{NE}}), m_n(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})) \\ &\geq V_n(f_n(\gamma_n, \boldsymbol{\gamma}_{-n}^{\mathrm{NE}}), \widetilde{r}(\gamma_n, \boldsymbol{\gamma}_{-n}^{\mathrm{NE}}), m_n(\gamma_n, \pi_n^{\mathrm{NE}}, \boldsymbol{\gamma}_{-n}^{\mathrm{NE}}, \boldsymbol{\pi}_{-n}^{\mathrm{NE}})), \\ &\qquad \gamma_n \in [0, \bar{r}], n \in \mathcal{N}. \end{aligned} \quad (13)$$

From (12) to (13), we replace $m_n(\gamma_n, \pi_n, \boldsymbol{\gamma}_{-n}^{\mathrm{NE}}, \boldsymbol{\pi}_{-n}^{\mathrm{NE}})$ with $m_n(\gamma_n, \pi_n^{\mathrm{NE}}, \boldsymbol{\gamma}_{-n}^{\mathrm{NE}}, \boldsymbol{\pi}_{-n}^{\mathrm{NE}})$. In the rest of this paper, we focus on the scenario where $N \geq 3$. When $N = 1$, cross-silo FL cannot be operated. When $N = 2$, we can introduce an additional virtual organization with zero utility and cost.

3.2.2 Nash Equilibrium and Properties P1–P3

According to Definition 2 and inequality (13), any NE should satisfy the following.

Lemma 1 (Nash Equilibrium)**.** *A message profile $(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$ is an NE of Game 1 if and only if for all $n \in \mathcal{N}$,*

$$\gamma_n^{\mathrm{NE}} = N \arg\max_{r\in[0,\bar{r}]} V_n(f_n^{\circ}(r), r, m_n(r\mathbf{1}, \boldsymbol{\pi}^{\mathrm{NE}})) - \sum_{n'\in\mathcal{N}\setminus\{n\}} \gamma_{n'}^{\mathrm{NE}}, \quad (14)$$

where $\mathbf{1}$ is an all-one vector with length N. The function $m_n(r\mathbf{1}, \boldsymbol{\pi}^{\mathrm{NE}}) = r(\pi_{\mu(n+1)}^{\mathrm{NE}} - \pi_{\mu(n+2)}^{\mathrm{NE}})$ defines the monetary transfer to organization n under r training rounds given $\boldsymbol{\pi}^{\mathrm{NE}}$.

The proof is given in Appendix A. Lemma 1 implies that $(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$ is an NE if and only if under $\boldsymbol{\pi}^{\mathrm{NE}}$, $\boldsymbol{\gamma}^{\mathrm{NE}}$ leads to the number of training rounds $\widetilde{r}(\boldsymbol{\gamma}^{\mathrm{NE}})$ that maximizes the payoff of each organization, i.e., $\sum_{n'\in\mathcal{N}} \gamma_{n'}^{\mathrm{NE}}/N = \widetilde{r}(\boldsymbol{\gamma}^{\mathrm{NE}}) = \arg\max_{r\in[0,\bar{r}]} V_n(f_n^{\circ}(r), r, m_n(r\mathbf{1}, \boldsymbol{\pi}^{\mathrm{NE}}))$, $n \in \mathcal{N}$.

According to Lemma 1 as well as equations (9)–(11), Mechanism 1 satisfies properties P1, P2, and P3.

Theorem 1 (Social Efficiency)**.** *Under any NE of Game 1, i.e., $(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$, the determined processing capacity $\boldsymbol{f}(\boldsymbol{\gamma}^{\mathrm{NE}})$ optimizes the nonconvex problem* (8)*.*

The proof of Theorem 1 is given in Appendix B.

Proposition 1 (Individual Rationality)**.** *Under any NE of Game 1, i.e. $(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$, each organization has nonnegative payoff, i.e., $V_n(f_n(\boldsymbol{\gamma}^{\mathrm{NE}}), \widetilde{r}(\boldsymbol{\gamma}^{\mathrm{NE}}), m_n(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})) \geq 0$, $n \in \mathcal{N}$.*

This holds because $V_n(f_n(\boldsymbol{\gamma}^{\rm NE}), \widetilde{r}(\boldsymbol{\gamma}^{\rm NE}), m_n(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})) \geq V_n(f_n^\circ(0), 0, m_n(\mathbf{0}, \boldsymbol{\pi}^{\rm NE})) = 0$ for $n \in \mathcal{N}$ due to Lemma 1, where $\mathbf{0}$ is a zero vector with length N.

Proposition 2 (Budget Balance)**.** *Under any NE of Game 1, i.e., $(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$, the summation of the monetary transfer of all organizations is equal to zero, i.e., $\sum_{n\in\mathcal{N}} m_n(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE}) = 0$.*

Proposition 2 is proven by substituting $m_n(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$ defined in (11) into $\sum_{n\in\mathcal{N}} m_n(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$.

3.2.3 Insights on Unit Monetary Transfer and Property P4

We now discuss the practical insights on the unit monetary transfer to the organizations. Recall that $\zeta_n^{\rm NE} \triangleq \zeta_n(\boldsymbol{\pi}^{\rm NE})$ denotes the unit monetary transfer of organization $n \in \mathcal{N}$ under an NE of Game 1. Let $r^{\rm NE} \triangleq \widetilde{r}(\boldsymbol{\gamma}^{\rm NE})$, and hence $\boldsymbol{f}^{\rm NE} \triangleq \boldsymbol{f}(\boldsymbol{\gamma}^{\rm NE}) = \boldsymbol{f}^\circ(r^{\rm NE})$. According to Lemma 1, the unit monetary transfer under NE satisfies the following.

Proposition 3 (Unit Monetary Transfer)**.** *If $r^{\rm NE} \in [0, \bar{r}]$, then*

$$\zeta_n^{\rm NE} = \frac{\partial C_n(f_n^\circ(r^{\rm NE}), r^{\rm NE}))}{\partial r^{\rm NE}} - \frac{\partial U_n(r^{\rm NE})}{\partial r^{\rm NE}}, n \in \mathcal{N}. \quad (15)$$

Proposition 3 holds because the partial derivative of $V_n(f_n^\circ(r), r, m_n(r\mathbf{1}, \boldsymbol{\pi}^{\rm NE}))$ with respect to r is equal to zero under $r = r^{\rm NE}$. Intuitively, the unit payment of each organization n (i.e., $-\zeta_n^{\rm NE}$) is equal to its marginal benefit.

Resulting from Proposition 3, no organization would prefer to deviate from the determined processing capacity under NE message profiles during the FL process.

Proposition 4. *For any organization $n \in \mathcal{N}$, given the unit monetary transfer $\zeta_n^{\rm NE}$ under an NE message profile,*

$$\boldsymbol{f}^{\rm NE} = \arg\max_{\boldsymbol{f}\in\mathbb{R}_+^N} V_n\left(f_n, r(\boldsymbol{f}), \zeta_n^{\rm NE} r(\boldsymbol{f}))\right). \quad (16)$$

Proposition 4 holds because $r^{\rm NE} = \sum_{n\in\mathcal{N}} \gamma_n/N$ maximizes the payoff of each organization based on Lemma 1, and $\boldsymbol{f}^{\rm NE} = \boldsymbol{f}^\circ(r^{\rm NE})$. The proof is given in Appendix C. Based on Proposition 4, Mechanism 1 satisfies property P4.

Corollary 1 (Fulfillment of Agreement)**.** *For any organization $n \in \mathcal{N}$, given the unit monetary transfer $\zeta_n^{\rm NE}$, we have $V_n(f_n^{\rm NE}, r(f_n^{\rm NE}, \boldsymbol{f}_{-n}^{\rm NE}), \zeta_n^{\rm NE} r(f_n^{\rm NE}, \boldsymbol{f}_{-n}^{\rm NE})) \geq V_n(f_n, r(f_n, \boldsymbol{f}_{-n}^{\rm NE}), \zeta_n^{\rm NE} r(f_n, \boldsymbol{f}_{-n}^{\rm NE}))$ for all $f_n \in \mathbb{R}_+$, $n \in \mathcal{N}$.*

Corollary 1 can be proven by contradiction. Suppose Corollary 1 does not hold. Then, Proposition 4 is violated. Corollary 1 implies that given the unit monetary transfer $\zeta_n^{\rm NE}$, each organization weakly prefers to use the processing capacity $\boldsymbol{f}(\boldsymbol{\gamma}^{\rm NE})$ determined by the NE.

Recall that $\boldsymbol{f}^*$ is the processing capacity that optimizes problem (8), and $\boldsymbol{f}_{-n}^* \triangleq (f_{n'}^*, n' \in \mathcal{N} \setminus \{n\})$. According to Theorem 1 and Corollary 1, our proposed incentive mechanism can prevent the free-rider attack.

Corollary 2 (Preventing Free-Rider Attack)**.** *Under Mechanism 1, given $\zeta_n^{\rm NE}$, no organization is better off by deviating from the optimal processing capacity $\boldsymbol{f}^*$ that optimizes problem* (8)*, i.e., $V_n(f_n^*, r(f_n^*, \boldsymbol{f}_{-n}^*), \zeta_n^{\rm NE} r(f_n^*, \boldsymbol{f}_{-n}^*)) \geq V_n(f_n, r(f_n, \boldsymbol{f}_{-n}^*), \zeta_n^{\rm NE} r(f_n, \boldsymbol{f}_{-n}^*))$ for all $f_n \in \mathbb{R}_+$, $n \in \mathcal{N}$.*

4 Blockchain-Empowered Framework

The incentive mechanism proposed in Section 3 satisfies properties P1−P4. However, with this mechanism, the organizations can determine their NE message profiles only when they know the private information (e.g., utility, cost) of each other. In this section, we propose a blockchain-empowered framework for deploying our proposed incentive mechanism and enabling the organizations to determine their NE without knowing the private information.

4.1 Overview of the Proposed Framework

Our proposed blockchain-empowered incentive mechanism framework is shown in Fig. 2. Specifically, organizations iteratively update and exchange their message profiles $\gamma_n(t)$ and $\pi_n(t)$ until convergence to NE through interacting with the blockchain, where $t \in \mathbb{Z}_+$ denotes the iteration index. The blockchain network includes (a) a smart contract for coordinating the message profile exchange between organizations, (b) a state that indicates the recent message profiles and convergence flags, and (c) a ledger for tracking the historical message profiles submitted by the organizations.

The smart contract is a self-executing digital agreement. It has three modules. The *message profile collection* module is responsible for gathering the message profiles $\gamma_n(t)$ and $\pi_n(t)$ of all organizations $n \in \mathcal{N}$ and sends $\boldsymbol{\gamma}(t) = (\gamma_n(t), n \in \mathcal{N})$ and $\boldsymbol{\pi}(t) = (\pi(t), n \in \mathcal{N})$ back to the organizations, based on which the organizations update their message profiles. The *convergence check* module is responsible for gathering the convergence flag, denoted by Convg_flag_n, of all organizations $n \in \mathcal{N}$. Based on the flags, this module determines whether the message profiles of all organizations have converged to NE or not and informs the organizations with the Global_convg_flag. If Convg_flag_n is True for all $n \in \mathcal{N}$, then Global_convg_flag is set to True. Otherwise, Global_convg_flag is set to False. If Global_convg_flag is True, then the convergence check module sends the NE message profiles $\boldsymbol{\gamma}^{\rm NE}$ and $\boldsymbol{\pi}^{\rm NE}$ to the organizations. Based on the monetary transfer $m_n(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$ for $n \in \mathcal{N}$ determined by (11), the *transaction minimization* module determines the monetary transfer $m_{n',n}(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$ from organization $n' \in \mathcal{N}$ to organization $n \in \mathcal{N} \setminus \{n'\}$ (see Section 4.3) and informs the organizations accordingly.

Recall that the protocol must be lightweight (Challenge 2 (ii)). Both the message profile collection and convergence check modules perform only simple executions (e.g., comparison, replacement) and have a computational complexity of $O(NI)$, where I denotes the number of iterations that the algorithm takes before convergence. Based on the numerical results in Section 5, I increases quadratically in N. We will show in Section 4.3 that the transaction minimization module has a complexity of $O(N^2 \log_2 N)$. Thus, all these modules have polynomial complexity.

In organization $n \in \mathcal{N}$, there are three modules. The *submit message profile* module sends the recent message profile $\gamma_n(t)$ and $\pi_n(t)$ to the blockchain and requests $\boldsymbol{\gamma}(t)$ and $\boldsymbol{\pi}(t)$. These profiles $\boldsymbol{\gamma}(t)$ and $\boldsymbol{\pi}(t)$ will be used for updating $\gamma_n(t+1)$ and $\pi_n(t+1)$ using *message profile update* module (see Section 4.2). The *convergence check* module is responsible for determining the recent Convg_flag_n and uploading it to the blockchain. If the Global_convg_flag sent

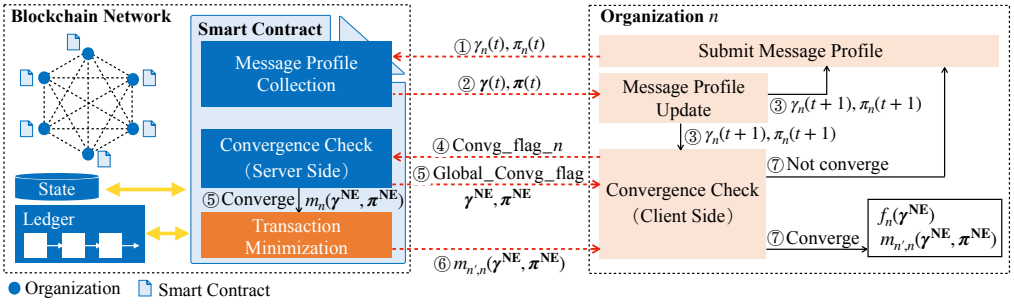


Fig. 2. An illustration of blockchain-empowered incentive mechanism framework.

by the blockchain is False, then organization n continues the iterative process and submits the updated message profile $\gamma_n(t+1)$ and $\pi_n(t+1)$. If Global_convg_flag is True, then the convergence check module determines the processing capacity $f_n(\boldsymbol{\gamma}^{\mathrm{NE}})$ based on $\boldsymbol{\gamma}^{\mathrm{NE}}$ and obtains the values of $m_{n',n}(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$ for $n' \in \mathcal{N}$ from the blockchain. Note that during the entire iterative process, organizations send only their message profiles and convergence indicators but not their private information to the blockchain (Challenge 2 (i)).

Under this framework, there are two remaining questions. First, how should each organization n update $\gamma_n(t+1)$ and $\pi_n(t+1)$ using the message profile update module such that the message profiles of all organizations can converge to an NE of Game 1? Second, how to determine the monetary transfer between each pair of organizations using the transaction minimization module? To answer these questions, we propose a distributed algorithm in Section 4.2, which defines the message profile update steps and the associated executions at each organization $n \in \mathcal{N}$ in Fig. 2. We propose an algorithm for transaction minimization in Section 4.3.

4.2 A Distributed Message Profile Update Algorithm

In our proposed framework, it is challenging to design an algorithm that enables organizations to iteratively update their message profiles based on their private information as well as $\boldsymbol{\gamma}_n(t)$ and $\boldsymbol{\pi}_n(t)$ such that the message profiles can gradually converge to the NE. This is due to the non-convexity of the social welfare maximization problem (8). To address this, we first reformulate problem (8). Since the saddle point of the Lagrangian of the reformulated problem is an NE of Game 1, we can then design an algorithm that converges to the saddle point of the Lagrangian and hence an NE of Game 1. We first reformulate the social welfare maximization problem. Then, we propose the algorithm.

4.2.1 Problem Reformulation

We introduce an auxiliary variable vector $\boldsymbol{r} = (r_n, n \in \mathcal{N})$, where r_n is the number of training rounds that organization n performs. Since r_n should be the same for all $n \in \mathcal{N}$, we have the following social welfare maximization problem:

$$\underset{\boldsymbol{r}}{\text{maximize}} \quad \sum_{n\in\mathcal{N}} \left(U_n\left(r_n\right) - C_n\left(f_n^{\circ}(r_n), r_n\right)\right) \tag{17a}$$

$$\text{subject to} \quad r_{\mu(n-2)} = r_{\mu(n-1)}, \ n \in \mathcal{N}, \tag{17b}$$

$$r_n \in [0, \bar{r}], \ n \in \mathcal{N}. \tag{17c}$$

Constraint (17b) can be expanded as $r_{N-2} = r_{N-1}$ for $n = 0$, $r_{N-1} = r_0$ for $n = 1$, and $r_{n-2} = r_{n-1}$ for $2 \le n \le N-1$. It implies that r_n has to be identical for all $n \in \mathcal{N}$. We write it in the form of $r_{\mu(n-2)} = r_{\mu(n-1)}, n \in \mathcal{N}$ for the simplicity of algorithm design. Let $\boldsymbol{r}^*$ be the optimal solution to problem (17), and let $r^* = r_0^* = \cdots = r_{N-1}^*$. Then, the processing capacity $\boldsymbol{f}^{\circ}(r^*) = (f_n^{\circ}(r^*), n \in \mathcal{N})$ optimizes problem (8).

We define the Lagrangian of problem (17), i.e., $\mathcal{L} : [0, \bar{r}]^N \times \mathbb{R}^N \to \mathbb{R}$, as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda}) = &\textstyle\sum_{n\in\mathcal{N}} \left(U_n\left(r_n\right) - C_n\left(f_n^{\circ}(r_n), r_n\right)\right) \\ &- \textstyle\sum_{n\in\mathcal{N}} \lambda_n \left(r_{\mu(n-2)} - r_{\mu(n-1)}\right),\end{aligned} \tag{18}$$

where $\boldsymbol{\lambda} = (\lambda_n, n \in \mathcal{N})$ is the vector of the Lagrange multipliers. Note that $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda})$ can be decoupled into multiple functions. That is, $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda}) = \sum_{n\in\mathcal{N}} \mathcal{L}_n(r_n, \boldsymbol{\lambda})$, where

$$\begin{aligned}\mathcal{L}_n(r_n, \boldsymbol{\lambda}) = &U_n\left(r_n\right) - C_n\left(f_n^{\circ}(r_n), r_n\right) \\ &- \left(\lambda_{\mu(n+2)} - \lambda_{\mu(n+1)}\right) r_n, \ n \in \mathcal{N}.\end{aligned} \tag{19}$$

For $n \in \mathcal{N}$, $\lambda_{\mu(n+2)}$ and $\lambda_{\mu(n+1)}$ are the Lagrange multipliers correspond to constraints $r_n = r_{\mu(n+1)}$ and $r_{\mu(n-1)} = r_n$, respectively. Based on (7), (10), (11), and (19), we have $\mathcal{L}_n(r_n, \boldsymbol{\lambda}) = V_n\left(f_n^{\circ}\left(r_n\right), r_n, m_n(r_n \mathbf{1}, \boldsymbol{\lambda})\right)$, i.e., the payoff of organization n under r_n training rounds given $\boldsymbol{\lambda}$.

For problem (17), strong duality holds according to Slater's condition. Hence, the saddle point of Lagrangian $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda})$, denoted by $(\boldsymbol{r}^*, \boldsymbol{\lambda}^*)$, exists. That is, $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda}^*) \le \mathcal{L}(\boldsymbol{r}^*, \boldsymbol{\lambda}^*) \le \mathcal{L}(\boldsymbol{r}^*, \boldsymbol{\lambda})$ for any $\boldsymbol{r} \in [0, \bar{r}]^N$, $\boldsymbol{\lambda} \in \mathbb{R}^N$. Thus, we can prove that if $(\boldsymbol{r}^*, \boldsymbol{\lambda}^*)$ is a saddle point of $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda})$, then it is an NE of Game 1, with the proof given in Appendix D.

Lemma 2 (Saddle Point and NE). *For any saddle point of $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda})$, denoted by $(\boldsymbol{r}^*, \boldsymbol{\lambda}^*)$, the message profile $(\boldsymbol{\gamma}^{\mathrm{NE}} = \boldsymbol{r}^*, \boldsymbol{\pi}^{\mathrm{NE}} = \boldsymbol{\lambda}^*)$ is an NE of Game 1.*

4.2.2 Algorithm Design

We propose the message profile update algorithm based on the distributed accelerated augmented Lagrangian method [49], which is a distributed algorithm for achieving the saddle point of the Lagrangian of a constrained problem and has a high convergence rate. We have modified the algorithm in [49] to adapt to the cross-silo FL scenario. In our proposed algorithm, we replace the notations $\boldsymbol{r}$ and $\boldsymbol{\lambda}$ in $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda})$ with notations $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$, respectively. The organizations aim to find a saddle point of $\mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\pi}) = \sum_{n\in\mathcal{N}} \mathcal{L}_n(\gamma_n, \boldsymbol{\pi}) = \sum_{n\in\mathcal{N}} V_n\left(f_n^{\circ}\left(\gamma_n\right), \gamma_n, m_n(\gamma_n \mathbf{1}, \boldsymbol{\pi})\right)$. The obtained saddle point is an NE of Game 1.

Our proposed algorithm is given in Algorithm 1, which specifies the detailed executions of the modules at each

Algorithm 1: Distributed Augmented Lagrangian-Based Message Profile Update Algorithm

1 Randomly initializes $\gamma_n(0)$, $\pi_n(0)$;
2 Convg_flag_$n \leftarrow$ False, $t \leftarrow 0$;
3 Send Convg_flag_n to the blockchain and wait for Global_convg_flag;
4 **while** *Global_convg_flag is False* **do**
5 Submit $(\gamma_n(t), \pi_n(t))$ and wait for the reply from the blockchain that contains $\boldsymbol{\gamma}(t)$ and $\boldsymbol{\pi}(t)$;
6 $\hat{\gamma}_n(t) \leftarrow \arg\max_{\gamma_n \in [0,\bar{r}]} V_n^\rho(\gamma_n, \boldsymbol{\gamma}_{-n}(t), \boldsymbol{\pi}(t))$;
7 $\gamma_n(t+1) \leftarrow \gamma_n(t) + \eta\,(\hat{\gamma}_n(t) - \gamma_n(t))$;
8 $\pi_n(t+1) \leftarrow \pi_n(t) + \rho\eta(\gamma_{\mu(n-2)}(t) - \gamma_{\mu(n-1)}(t))$;
9 **if** $|\gamma_n(t+1) - \gamma_n(t)| \le \phi$ **then**
10 Convg_flag_$n \leftarrow$ True;
11 **end**
12 Send Convg_flag_n to the blockchain and wait for Global_convg_flag, $t \leftarrow t+1$;
13 **end**

organization $n \in \mathcal{N}$ in Fig. 2. Specifically, the organizations update the message profiles for multiple iterations using the blockchain network until convergence. Each organization n first randomly initializes its message profile $(\gamma_n(0), \pi_n(0))$. While the convergence indicator Global_convg_flag is False, each organization n submits $(\gamma_n(t), \pi_n(t))$ to the blockchain in iteration t. Then, the organizations wait for the reply from the blockchain, which contains the message profiles of all organizations $(\boldsymbol{\gamma}(t), \boldsymbol{\pi}(t))$. Steps 6–8 define how each organization n updates its message profile in the message profile update module. In Step 6, organization n computes $\hat{\gamma}_n(t)$ by deriving the value of $\gamma_n \in [0, \bar{r}]$ that maximizes

$$V_n^\rho(\gamma_n, \boldsymbol{\gamma}_{-n}, \boldsymbol{\pi}) = V_n(f_n^\circ(\gamma_n), \gamma_n, m_n(\gamma_n \mathbf{1}, \boldsymbol{\pi})) - \rho \textstyle\sum_{n \in \mathcal{N}} \left(\gamma_{\mu(n-2)} - \gamma_{\mu(n-1)}\right)^2, \quad (20)$$

where ρ is a penalty coefficient. The second term can be regarded as a term for penalizing the different number of training rounds submitted by the organizations. In Steps 7 and 8, organization n computes the updated message profile $(\gamma_n(t+1), \pi_n(t+1))$ by considering a step size $\eta \in (0, 1)$. A larger η implies a more aggressive update. An intuition behind Step 8 is as follows. Suppose the number of training rounds submitted by organization $\mu(n-2)$ is much larger than that submitted by organization $\mu(n-1)$ (i.e., the difference $\gamma_{\mu(n-2)}(t) - \gamma_{\mu(n-1)}(t)$ is large). Then, $\pi_n(t+1)$ is large based on Step 8. According to (10) and (11), the large value of $\pi_n(t+1)$ will lead to a small monetary transfer to organization $\mu(n-2)$ and a large monetary transfer to organization $\mu(n-1)$. Hence, in the next iteration, organizations $\mu(n-2)$ and $\mu(n-1)$ will reduce and increase the number of training rounds that they submit, respectively.

The algorithm terminates when the absolute difference between $\gamma_n(t+1)$ and $\gamma_n(t)$ for all $n \in \mathcal{N}$ is smaller than a predefined threshold ϕ. Thus, the organizations check whether their values of $|\gamma_n(t+1) - \gamma_n(t)|$ is smaller than threshold ϕ or not. They send their Convg_flag_n to the blockchain and wait for the Global_convg_flag replied by the blockchain. When the threshold ϕ is sufficiently small, Algorithm 1 is proven to converge to the NE of Game 1. This can be proven based on [49, Theorem 2] and Lemma 2.

Proposition 5 (Convergence)**.** *For any $\rho \in \mathbb{R}_+$ that ensures $V_n^\rho(\gamma_n, \boldsymbol{\gamma}_{-n}, \boldsymbol{\pi})$ to be strictly concave for $\boldsymbol{\gamma} \in [0, \bar{r}]^N$, $\boldsymbol{\pi} \in \mathbb{R}^N$, $n \in \mathcal{N}$, Algorithm 1 converges to the NE of Game 1.*

Recall that I denotes the number of iterations that Algorithm 1 takes to converge. Suppose a gradient decent (GD) algorithm is used for computing $\hat{\gamma}_n(t)$ in Step 6, and suppose this GD algorithm terminates when it finds a γ_n that improves function $V_n^\rho(\gamma_n, \boldsymbol{\gamma}_{-n}(t), \boldsymbol{\pi}(t))$ by less than σ.

Lemma 3 (Complexity)**.** *If a GD algorithm is used for computing $\hat{\gamma}_n(t)$ in Step 6, then the computational complexity of Algorithm 1 executed at organization $n \in \mathcal{N}$ is $\mathcal{O}(I \log(1/\sigma) + I)$.*

Lemma 3 holds because $V_n^\rho(\gamma_n, \boldsymbol{\gamma}_{-n}(t), \boldsymbol{\pi}(t))$ is concave in γ_n, under which the complexity of a GD algorithm for computing $\hat{\gamma}_n(t)$ in Step 6 is $\log(1/\sigma)$, and the executions in Steps 7–12 take a constant number of operations. In Section 5, we show numerically that I is around $\mathcal{O}(N^2)$, under which Algorithm 1 has a polynomial complexity.

4.3 Transaction Minimization

When Algorithm 1 has converged to the NE message profiles $\boldsymbol{\gamma}^{\mathrm{NE}}$ and $\boldsymbol{\pi}^{\mathrm{NE}}$, organization n can determine its processing capacity $f_n(\boldsymbol{\gamma}^{\mathrm{NE}})$ based on (9) and the monetary transfer $m_n(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$ based on (11). However, an arbitrary organization n' still needs to determine the exact payment to another organization n, denoted by $m_{n',n}(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}}) \ge 0$. In the blockchain network, each transaction between two organizations incurs a transaction fee. Thus, the transaction minimization module in the smart contract needs to determine the values of $m_{n',n}(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$ in order to minimize their total payment for transactions (Challenge 3).

We use m_n^{NE} to denote $m_n(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$. We use $m_{n',n}$ to denote $m_{n',n}(\boldsymbol{\gamma}^{\mathrm{NE}}, \boldsymbol{\pi}^{\mathrm{NE}})$, which is a decision variable. Let $\mathcal{N}^+$ and $\mathcal{N}^-$ denote the set of organizations with nonnegative and negative m_n^{NE}, respectively. The organizations aim to minimize the number of transactions:

$$\begin{aligned}
\underset{\boldsymbol{m}}{\text{minimize}} \quad & \textstyle\sum_{n \in \mathcal{N}} \sum_{n' \in \mathcal{N}} \mathbf{1}\,(m_{n',n} > 0) && \text{(21a)}\\
\text{subject to} \quad & \textstyle\sum_{n' \in \mathcal{N}} m_{n',n} = m_n^{\mathrm{NE}},\ n \in \mathcal{N}^+, && \text{(21b)}\\
& \textstyle\sum_{n' \in \mathcal{N}} m_{n,n'} = |m_n^{\mathrm{NE}}|,\ n \in \mathcal{N}^-, && \text{(21c)}\\
& m_{n',n} \ge 0,\ n', n \in \mathcal{N}. && \text{(21d)}
\end{aligned}$$

In problem (21), the objective is to minimize the number of positive $m_{n',n}$, under which a transaction fee is incurred. Constraint (21b) ensures that each organization $n \in \mathcal{N}^+$ gets paid m_n^{NE}, and constraint (21c) ensures that each organization $n \in \mathcal{N}^-$ pays a total of m_n^{NE} to other organizations. Problem (21) is NP-hard. This is because even if we restrict m_n^{NE} to be integers, this problem can be reduced to the subset sum problem [50], which is a typical NP-hard problem.

4.3.1 Transaction Minimization Algorithm

To solve problem (21), we propose Algorithm 2, which will be performed by the transaction minimization module in the smart contract. At the beginning of Algorithm 2, we initialize $\boldsymbol{M}^+$ and $\boldsymbol{M}^-$ as the vectors with nonnegative and negative monetary transfer, respectively. Let M_i^+ and

Algorithm 2: Transaction Minimization

1 $\boldsymbol{M}^+ \leftarrow (m_n^{\mathrm{NE}}, n \in \mathcal{N}^+)$, $\boldsymbol{M}^- \leftarrow (m_n^{\mathrm{NE}}, n \in \mathcal{N}^-)$;
2 $m_{n',n} \leftarrow 0$ for $n', n \in \mathcal{N}$;
3 **while** $\boldsymbol{M}^+ \neq \mathbf{0}$ *and* $\boldsymbol{M}^- \neq \mathbf{0}$ **do**
4 **while** *Set* $\mathcal{C} \triangleq \{(i,j) \mid M_i^+ = -M_j^-, M_i^+ \neq 0\}$ *is nonempty* **do**
5 $(i,j) \leftarrow$ an arbitrary element in $\mathcal{C}$;
6 $m_{n^-(j),n^+(i)} \leftarrow M_i^+$;
7 $M_i^+ \leftarrow 0$, $M_j^- \leftarrow 0$;
8 **end**
9 $i \leftarrow \arg\min_{i'}\{|M_{i'}^+| \mid M_{i'}^+ \neq 0, i' \in \mathcal{N}^+\}$;
10 $j \leftarrow \arg\min_{j'}\{|M_{j'}^-| \mid M_{j'}^- \neq 0, j' \in \mathcal{N}^-\}$;
11 $m_{n^-(j),n^+(i)} \leftarrow \min\{M_i^+, M_j^-\}$;
12 $M_i^+ \leftarrow M_i^+ - m_{n^-(j),n^+(i)}$;
13 $M_j^- \leftarrow M_j^- + m_{n^-(j),n^+(i)}$;
14 **end**

M_j^- denote the i^{th} element in vector $\boldsymbol{M}^+$ and j^{th} element in vector $\boldsymbol{M}^-$, respectively. Let $n^+(i)$ and $n^-(j)$ denote the organization indices associated with M_i^+ and M_j^-, respectively. The values of M_i^+ and M_j^- will be updated across iterations. They represent the remaining monetary transfer of the corresponding organizations that have not been assigned to pay or receive by those transactions $m_{n',n}$ for $n', n \in \mathcal{N}$ which have already been determined.

The iteration continues as long as vectors $\boldsymbol{M}^+$ and $\boldsymbol{M}^-$ are nonzero. Note that exactly one of $\boldsymbol{M}^+$ and $\boldsymbol{M}^-$ being a nonzero vector cannot happen, because $\sum_{n\in\mathcal{N}} m_n^{\mathrm{NE}} = 0$ holds based on Proposition 2. In steps 4−8, we find pairs of elements in $\boldsymbol{M}^+$ and $\boldsymbol{M}^-$ such that $M_i^+ = -M_j^-$. These are the element pairs that can be canceled out by letting organization $n^-(j)$ compensate organization $n^+(i)$ with payment M_i^+. If no such pair exists, then in steps 9−10, we find the indices of M_i^+ and M_j^- which have the minimum nonzero absolute value. In steps 11−13, organization $n^-(j)$ compensates organization $n^+(i)$ with payment $\min\{M_i^+, M_j^-\}$. The values of M_i^+ and M_j^- are updated accordingly. Through steps 9−13, in each iteration, at least one element in vectors $\boldsymbol{M}^+$ and $\boldsymbol{M}^-$ can be reduced to zero.

Let L^* denote the minimum number of transactions, i.e., the optimal value to problem (21). Let L° denote the number of transactions obtained using Algorithm 2. We show that the output of Algorithm 2 is no larger than twice the minimum number of transactions, and it has a polynomial complexity. Propositions 6 and 7 are proven in Appendices E and F, respectively.

Proposition 6 (Competitive ratio)**.** *Algorithm 2 achieves a competitive ratio of two, i.e.,* $L^\circ/L^* \leq 2$.

Proposition 7 (Complexity)**.** *Algorithm 2 has a computational complexity of* $O(N^2 \log_2 N)$.

5 Performance Evaluation

We first conduct simulations to evaluate the performance of our proposed Mechanism 1, Algorithm 1, and Algorithm 2. Then, we implement our proposed framework with Ethereum [34] for real-world evaluation. Table 1 shows the parameter settings. We perform FL using the MNIST dataset with convolutional neural network model, based on which we obtain parameters S_n, D_n, model size (i.e., the size of $\boldsymbol{w}$), ϵ_0, and ϵ_1. In the simulations, MNIST dataset is randomly distributed to organizations.[4] The values of ϵ_0 and ϵ_1 are determined by fitting the FL precision using function (4).

TABLE 1
List of Parameters

Par.	Value	Par.	Value
N	10	Model size	0.16 Mbits
K	5	DL speed	78.26 Mbps [51]
T	60 seconds	UL speed	42.06 Mbps [51]
D_n	0.01 gigacycles	Invest. cost	\$0.22 per GHz per hour [52]
S_n	600 samples	DL energy	3 Joules per Mbit [53]
ϵ_0	9.82	UL energy	3 Joules per Mbit [53]
ϵ_1	4.26	Elec. rate	\$0.174 per kWh [54]

5.1 Performance of Mechanism 1 and Algorithm 1

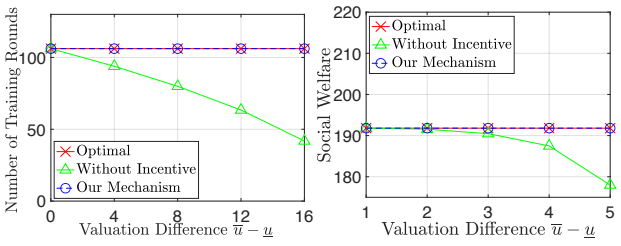
We conduct simulations to show the performance of our proposed Mechanism 1, denoted by "our mechanism". Algorithm 1 is used to determine the output of Mechanism 1. The performance is compared with the optimal value of problem (8), denoted by "optimal", and the performance without any incentive mechanism (denoted by "without incentive"). We consider a scenario where organizations have heterogeneous unit revenue (i.e., heterogeneous valuation over the global model): $u_n = \underline{u}$ for $n \in \{0, 1, ..., \lfloor N/2 \rfloor - 1\}$ and $u_n = \overline{u}$ for $n \in \{\lfloor N/2 \rfloor, \lfloor N/2 \rfloor + 1, ..., N-1\}$, where the average unit revenue is 10.[5] Fig. 3 shows that our proposed mechanism always leads to the same performance (i.e., the number of training rounds $r(\boldsymbol{f})$ in FL process and social welfare) as the optimal solution, which validates Theorem 1. When compared with the scenario of "without incentive", our proposed mechanism improves the number of training rounds that organizations perform in FL process and improves the social welfare, which validates its capability in preventing the free-rider attack. The improvement is more significant when the valuation difference is higher.

We evaluate the convergence of our proposed Algorithm 1 (denoted by "Augmented Lagrangian") and another Lagrangian-based algorithm [46, Section 8.1],[6] denoted by "Lagrangian". Note that both algorithms can be used to determine the output of Mechanism 1 and converge to the NE of Game 1. We conduct 100 simulation rounds. In each round, the values of u_n for $n \in \mathcal{N}$ are randomly generated using the truncated normal distribution with a mean of 10 and a standard deviation of one. The results are shown using boxplot. In Fig. 4(a), our proposed algorithm reduces the number of iterations by $38.7\% - 69.9\%$. The reduction is more significant when the number of organizations is large. Fig. 4(b) shows that as the number of organizations

4. This corresponds to the scenario where organizations have IID datasets. The non-IID scenario affects only the values of ϵ_0, and ϵ_1, while the practical insights (e.g., in terms of algorithm convergence, social welfare improvement) remain unchanged.

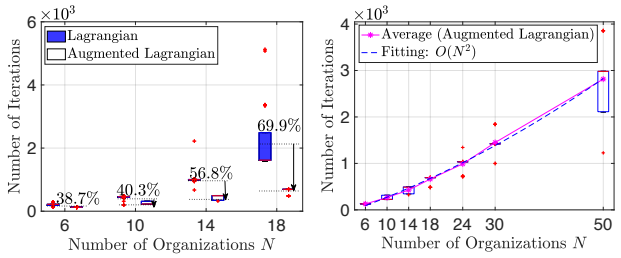
5. The NE under $u_n = 10$ for $n \in \mathcal{N}$ leads to the number of training rounds that can achieve a digit recognition correctness rate of more than 90%. Our observations also hold for other values of the mean.

6. The Lagrangian-based algorithm [46, Section 8.1] is an algorithm for finding the saddle point of any Lagrangian. In the simulation, it has been modified to solve the problem in the cross-silo FL scenario.



(a) (b)

Fig. 3. Comparison between the optimal solution to problem (8), performance without incentive mechanism, and with our proposed incentive mechanism: (a) number of training rounds in FL; (b) social welfare.



(a) (b)

Fig. 4. (a) Comparison between the benchmark method "Lagrangian" and Algorithm 1, denoted by "Augmented Lagrangian". The red solid and black dash lines show the corresponding median and mean values, respectively. (b) Empirical computational complexity of Algorithm 1.

increases, the number of iterations that our proposed algorithm takes has a polynomial complexity.

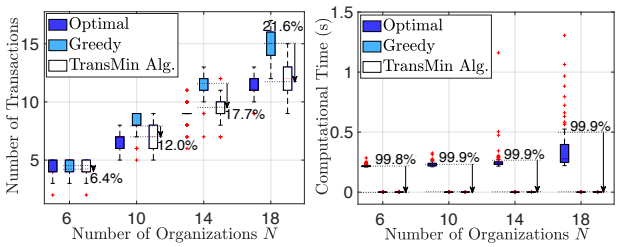
5.2 Performance of Algorithm 2

We conduct simulations to evaluate the performance of Algorithm 2 (denoted by "TransMin alg.") for transaction minimization. The results are compared with a greedy algorithm, where the organizations with negative monetary transfer greedily compensate the organizations with positive monetary transfer in sequence (according to the order of their indices), and the optimal solution to problem (21). We run simulations for a hundred rounds. In each round, each monetary transfer between the organizations is randomized with uniform distribution within the range $[-1, 1]$.[7] Fig. 5(a) shows that our proposed algorithm achieves a comparable performance with the optimal solution. When compared with the greedy algorithm, our proposed algorithm reduces the number of transactions by up to 21.6%. In Fig. 5(b), when compared with the optimal solution, our algorithm reduces the computational time for determining the transactions between organizations by up to 99.9%.

5.3 Real-World Implementation with Ethereum

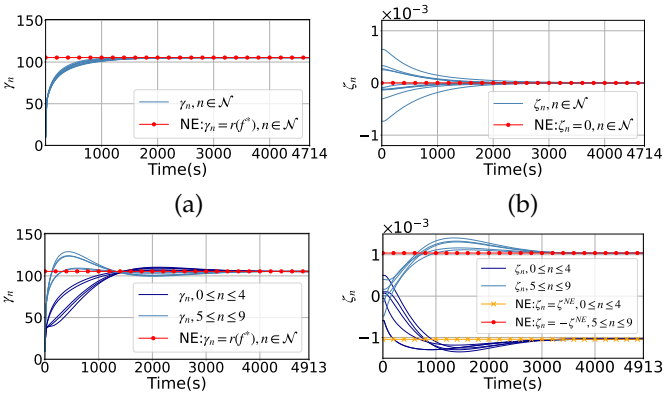
We build a testbed using Ethereum to evaluate the performance of our proposed blockchain-empowered framework. The smart contract is written in Solidity, and the programs at organizations are written in Python. In the following experiments, we consider two scenarios: a homogeneous scenario where all organizations have the same unit revenue $u_n = 10$; a heterogeneous scenario where $u_n = 4$ for organizations $n \in \{0, 1, ..., \lfloor N/2 \rfloor - 1\}$ and $u_n = 16$ for $n \in \{\lfloor N/2 \rfloor, \lfloor N/2 \rfloor + 1, ..., N-1\}$.

7. Such a distribution range does not lose the generality of the simulation results, because only the relative monetary transfer (rather than the actual values) between the organizations affects the performance.



(a) (b)

Fig. 5. Comparison between the optimal solution to problem (21), a greedy algorithm, and Algorithm 2 (denoted by TransMin Alg.): (a) number of transactions; (b) computational time. The black dash lines show the corresponding mean values.



(a) (b) (c) (d)

Fig. 6. Convergence of the message profiles ($N = 10$) in the testbed: (a) γ_n in homogeneous scenario; (b) ζ_n in homogeneous scenario; (c) γ_n in heterogeneous scenario; (d) ζ_n in heterogeneous scenario.

Fig. 6 shows the convergence of the message profiles submitted by the organizations across wall-clock time. Figs. 6(a) and 6(b) correspond to the homogeneous scenario. As the time increases, γ_n gradually converges to the optimal number of training rounds $r(\boldsymbol{f}^*)$ for all $n \in \mathcal{N}$, where $\boldsymbol{f}^*$ is the optimal solution to problem (8). This implies that after the algorithm has converged, the social welfare is maximized. In addition, the unit monetary transfer $\zeta_n \triangleq \pi_{\mu(n+1)} - \pi_{\mu(n+2)}$ converges to zero for all $n \in \mathcal{N}$. Intuitively, since the organizations are homogeneous, they do not need to pay each other to motivate cooperation.

Figs. 6(c) and 6(d) correspond to the heterogeneous scenario. As the time increases, γ_n gradually converges to $r(\boldsymbol{f}^*)$. For the unit monetary transfer, ζ_n for $n \in \{0, 1, ..., 4\}$ converges to a positive value ζ^{NE}, and ζ_n for $n \in \{5, 6, ..., 9\}$ converges to the negative value of ζ^{NE}, i.e., $-\zeta^{\text{NE}}$. Note that ζ^{NE} and $-\zeta^{\text{NE}}$ are the unit monetary transfer for an organization $n \in \{0, 1, ..., 4\}$ and $n \in \{5, 6, ..., 9\}$ under NE, respectively. Intuitively, organizations in set $\{0, 1, ..., 4\}$ have a lower valuation, so the other organizations have to pay them to motivate their participation.

Fig. 7 shows the convergence of Algorithm 1 under values of N in the testbed. In Fig. 7(a), as N increases from 6 to 12, an additional organization leads to approximately 12.6 more iterations on average. Meanwhile, the unit revenue of the organizations (i.e., homogeneous or heterogeneous) does not affect the convergence. In Fig. 7(b), as N increases,

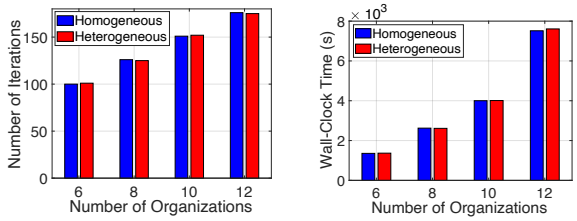


Fig. 7. Convergence under different number of organizations: (a) number of iterations; (b) wall-clock time.

the time that Algorithm 1 takes until convergence increases, and the marginal increase is non-decreasing. This is because when there are more organizations submitting messages, the blockchain system needs a longer time to generate blocks and to update the state and ledger. When N is equal to 10, the organizations require 1.11 hours to reach an agreement in terms of processing capacity and monetary transfer. This duration is relatively short when compared with the duration of the FL process (e.g., 1-10 days [2]).

6 Conclusion

In this work, we proposed a blockchain-empowered incentive mechanism framework in order to motivate efficient cooperation of the organizations in terms of their choices of processing capacities in cross-silo FL. This framework prevents the free-rider attack and enables organizations to achieve social efficiency, individual rationality, budget balance, and fulfillment of agreement without any central entity and the private information of the organizations. Simulation results verify that our proposed distributed message profile update algorithm converges to social efficiency solution and has a faster convergence than the conventional Lagrangian method Thus, this algorithm reduces the computational load in the smart contract. Our proposed transaction minimization algorithm empirically achieves a near-optimal solution with polynomial complexity. We implemented our proposed framework in a testbed and showed that it takes tens of minutes for our proposed algorithm to converge to the NE in real-world blockchain systems. One future direction is to incentivize efficient participation by jointly considering the computational resource, the number of local epochs, and the number of data samples used for training. Another direction is to allow organizations to join in only some of the training rounds based on their valuations.

References

[1] M. Tang and V.W.S. Wong, "An incentive mechanism for cross-silo federated learning: A public goods perspective," in *Proc. IEEE INFOCOM*, May 2021.

[2] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1, Mar. 2021.

[3] J. Sun, Y. Wu, S. Wang, Y. Fu, and X. Chang, "Permissioned blockchain frame for secure federated learning," *IEEE Commun. Lett.*, vol. 26, no. 1, pp. 13–17, Jan. 2022.

[4] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *Proc. IEEE INFOCOM*, May 2021.

[5] Y. Tao, Y. Jiang, F.-C. Zheng, Z. Wang, P. Zhu, M. Tao, D. Niyato, and X. You, "Content popularity prediction based on quantized federated Bayesian learning in fog radio access networks," *IEEE Trans. Commun.*, vol. 71, no. 2, pp. 893–907, Feb. 2023.

[6] Owkin, "Understanding complex biology through ai," https://www.owkin.com/#data-connect, accessed Jan. 30, 2024.

[7] MELLODDY, "Machine learning ledger orchestration for drug discovery," https://www.melloddy.eu, accessed Jan. 30, 2024.

[8] Intellegens, "Ichnite," https://intellegens.com/products-services/ichnite/, accessed Jan. 30, 2024.

[9] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int'l Conf. on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, FL, Apr. 2017.

[10] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. Int'l Conf. on Machine Learning (ICML)*, Jul. 2020.

[11] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int'l Conf. on Learning Representations (ICLR)*, Apr. 2020.

[12] X. Liu, Z. Zhong, Y. Zhou, D. Wu, X. Chen, M. Chen, and Q. Z. Sheng, "Accelerating federated learning via parallel servers: A theoretically guaranteed approach," *IEEE/ACM Trans. Netw.*, vol. 30, no. 5, pp. 2201–2215, Oct. 2022.

[13] S. Hosseinalipour, S. S. Azam, C. G. Brinton, N. Michelusi, V. Aggarwal, D. J. Love, and H. Dai, "Multi-stage hybrid federated learning over large-scale D2D-enabled fog networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1569–1584, Aug. 2022.

[14] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[15] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys & Tuts.*, vol. 22, no. 3, pp. 2031–2063, Third Quarter 2020.

[16] X. Wang, X. Ren, C. Qiu, Z. Xiong, H. Yao, and V.C.M. Leung, "Integrating edge intelligence and blockchain: What, why, and how," *IEEE Commun. Surveys & Tuts.*, vol. 24, no. 4, pp. 2193–2229, Fourth Quarter 2022.

[17] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12 806–12 825, Aug. 2021.

[18] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

[19] H. Jin, X. Dai, J. Xiao, B. Li, H. Li, and Y. Zhang, "Cross-cluster federated learning and blockchain for Internet of medical things," *IEEE Internet of Things J.*, vol. 8, no. 21, pp. 15 776–15 784, Nov. 2021.

[20] S. Otoum, I. A. Ridhawi, and H. Mouftah, "Securing critical IoT infrastructures with blockchain-supported federated learning," *IEEE Internet of Things J.*, vol. 9, no. 4, pp. 2592–2601, Feb. 2022.

[21] R. Cornes and T. Sandler, *The Theory of Externalities, Public Goods, and Club Goods*. Cambridge University Press, 1996.

[22] 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.

[23] J. Weng, J. Weng, H. Huang, C. Cai, and C. Wang, "FedServing: A federated prediction serving framework based on incentive mechanism," in *Proc. IEEE INFOCOM*, May 2021.

[24] J. S. Ng, W. Y. B. Lim, Z. Xiong, X. Cao, D. Niyato, C. Leung, and D. I. Kim, "A hierarchical incentive design toward motivating participation in coded federated learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 359–375, Jan. 2022.

[25] B. Zhao, X. Liu, W.-N. Chen, and R. Deng, "CrowdFL: Privacy-preserving mobile crowdsensing system via federated learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 8, pp. 4607–4619, Aug. 2023.

[26] Y. Yuan, L. Jiao, K. Zhu, and L. Zhang, "Incentivizing federated learning under long-term energy constraint via online randomized auctions," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5129–5144, Jul. 2022.

[27] J. Pang, J. Yu, R. Zhou, and J. C. Lui, "An incentive auction for heterogeneous client selection in federated learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5733–5750, Oct. 2023.

[28] P. Sun, H. Che, Z. Wang, Y. Wang, T. Wang, L. Wu, and H. Shao, "Pain-FL: Personalized privacy-preserving incentive for federated

learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3805–3820, Dec. 2021.
[29] 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, Jan. 2021.
[30] T. Zeng, O. Semiariy, M. Chen, W. Saad, and M. Bennis, "Federated learning on the road autonomous controller design for connected and autonomous vehicles," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 10 407–10 423, Dec. 2022.
[31] A. Kakhbod and D. Teneketzis, "An efficient game form for multirate multicast service provisioning," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 11, pp. 2093–2104, Dec. 2012.
[32] M. Zhang, J. Huang, and R. Zhang, "Wireless power transfer with information asymmetry: A public goods perspective," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 276–291, Jan. 2021.
[33] M. Tang, H. Pang, S. Wang, L. Gao, J. Huang, and L. Sun, "Multi-dimensional auction mechanisms for crowdsourced mobile video streaming," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2062–2075, Oct. 2018.
[34] Ethereum, "Welcome to Ethereum," https://ethereum.org/en/, accessed Jan. 30, 2024.
[35] X. Wu and H. Yu, "MarS-FL: Enabling competitors to collaborate in federated learning," *IEEE Trans. Big Data*, Jun. 2022 (Early Access).
[36] M. Zhang, E. Wei, and R. Berry, "Faithful edge federated learning: Scalability and privacy," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3790–3804, Dec. 2021.
[37] N. Zhang, Q. Ma, and X. Chen, "Enabling long-term cooperation in cross-silo federated learning: A repeated game perspective," *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 3910–3924, Jul. 2023.
[38] G. Huang, X. Chen, T. Ouyang, Q. Ma, L. Chen, and J. Zhang, "Collaboration in participant-centric federated learning: A game-theoretical perspective," *IEEE Trans. Mobile Comput.*, vol. 22, no. 11, pp. 6311–6326, Nov. 2023.
[39] 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 of Things J.*, vol. 7, no. 10, pp. 9575–9588, Oct. 2020.
[40] F. Luo, S. Al-Kuwari, and Y. Ding, "SVFL: Efficient secure aggregation and verification for cross-silo federated learning," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 850–864, Jan. 2024.
[41] H. Li, Y. Sun, Y. Yu, D. Li, Z. Guan, and J. Liu, "Privacy-preserving cross-silo federated learning atop blockchain for IoT," *IEEE Internet of Things J.*, vol. 10, no. 24, pp. 21 176–21 186, Dec. 2023.
[42] C. Zhang, Y. Xu, H. Elahi, D. Zhang, Y. Tan, J. Chen, and Y. Zhang, "A blockchain-based model migration approach for secure and sustainable federated learning in IoT systems," *IEEE Internet of Things J.*, vol. 10, no. 8, pp. 6774–6585, Apr. 2023.
[43] S. Fan, H. Zhang, Y. Zeng, and W. Cai, "Hybrid blockchain-based resource trading system for federated learning in edge computing," *IEEE Internet of Things J.*, vol. 8, no. 4, pp. 2252–2264, Feb. 2021.
[44] 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 of Things J.*, vol. 6, no. 6, pp. 10 700–10 714, Dec. 2019.
[45] B. Chen, H. Zeng, T. Xiang, S. Guo, T. Zhang, and Y. Liu, "ESB-FL: Efficient and secure blockchain-based federated learning with fair payment," *IEEE Trans. Big Data*, May 2022 (Early Access).
[46] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numerica*, vol. 14, pp. 1–137, 2005.
[47] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7947–7962, Dec. 2021.
[48] Microsoft Azure, "Linux virtual machines pricing," https://azure.microsoft.com/en-ca/pricing/details/virtual-machines/linux/, accessed Jan. 30, 2024.
[49] N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos, "An augmented Lagrangian method for distributed optimization," *Mathematical Programming*, vol. 152, pp. 405–434, Sep. 2015.
[50] J. Erickson, "NP-hard problems," https://jeffe.cs.illinois.edu/teaching/algorithms/book/12-nphard.pdf, accessed Jan. 30, 2024.
[51] SPEEDTEST, "Speedtest global index," https://www.speedtest.net/global-index, accessed Jan. 30, 2024.
[52] Microsoft Azure, "Cloud services pricing," https://azure.microsoft.com/en-us/pricing/details/cloud-services, accessed Jan. 30, 2024.
[53] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: A measurement study and implications for network applications," in *Proc. Internet Measurement Conf.*, Chicago, IL, Nov. 2009.
[54] E. Hub, "Electricity prices in Canada 2020," https://energyhub.org/wp-content/uploads/Electricity-Prices-in-Canada-2020.pdf, accessed Jan. 30, 2024.

Ming Tang (S'16, M'18) received her Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, China, in Sep. 2018. She worked as a postdoctoral research fellow at The University of British Columbia, Vancouver, Canada, from Nov. 2018 to Jan. 2022. Currently, she is an Associate Professor in the Department of Computer Science and Engineering and the Research Institute of Trustworthy Autonomous Systems at the Southern University of Science and Technology, Shenzhen, China. She received the 2022 Best Paper Award from *IEEE Transactions on Mobile Computing*.

Fu Peng is currently working toward the PhD degree in the Department of Computer Science and Technology, The Southern University of Science and Technology (SUSTech). His research interests include federated learning, neural network quantization, and evolutionary algorithms.

Vincent W.S. Wong (S'94, M'00, SM'07, F'16) received the B.Sc. degree from the University of Manitoba, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. (now Microchip Technology Inc.). He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently a Professor. His research areas include protocol design, optimization, and resource management of communication networks, with applications to 5G/6G wireless networks, Internet of things, mobile edge computing, smart grid, and energy systems. Dr. Wong is the Editor-in-Chief of *IEEE Transactions on Wireless Communications*. He has served as an Area Editor of *IEEE Transactions on Communications* and *IEEE Open Journal of the Communications Society*, an Associate Editor of *IEEE Transactions on Mobile Computing* and *IEEE Transactions on Vehicular Technology*, and a Guest Editor of *IEEE Journal on Selected Areas in Communications*, *IEEE Internet of Things Journal*, and *IEEE Wireless Communications*. Dr. Wong is a General Chair of *IEEE INFOCOM* 2024. He was a Tutorial Co-Chair of *IEEE GLOBECOM*'18, a Technical Program Co-chair of *IEEE VTC*2020-*Fall* and *IEEE SmartGridComm*'14, and a Symposium Co-chair of *IEEE ICC*'18, *IEEE SmartGridComm* ('13, '17) and *IEEE GLOBECOM*'13. He received the 2022 Best Paper Award from *IEEE Transactions on Mobile Computing*, and Best Paper Award at the *IEEE ICC* 2022 and *IEEE GLOBECOM* 2020. He has served as the Chair of the IEEE Vancouver Joint Communications Chapter and IEEE Communications Society Emerging Technical Sub-Committee on Smart Grid Communications. Dr. Wong is an IEEE Vehicular Technology Society Distinguished Lecturer (2023−2025) and was an IEEE Communications Society Distinguished Lecturer (2019−2020).

APPENDIX A
PROOF FOR LEMMA 1

Suppose $(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$ satisfies (14) but is not an NE. Then, there exists γ_n' for $n \in \mathcal{N}$ such that for $\boldsymbol{\gamma}' = (\gamma_n', \boldsymbol{\gamma}_{-n}^{\rm NE})$,

$$V_n(f_n^\circ(\widetilde{r}(\boldsymbol{\gamma}')), \widetilde{r}(\boldsymbol{\gamma}'), m_n(\widetilde{r}(\boldsymbol{\gamma}')\mathbf{1}, \boldsymbol{\pi}^{\rm NE})) > V_n(f_n^\circ(\widetilde{r}(\boldsymbol{\gamma}^{\rm NE})), \widetilde{r}(\boldsymbol{\gamma}^{\rm NE}), m_n(\widetilde{r}(\boldsymbol{\gamma}^{\rm NE})\mathbf{1}, \boldsymbol{\pi}^{\rm NE})), \quad (22)$$

where inequality (22) contradicts inequality (14).

Suppose $(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$ is an NE of Game 1, while (14) does not hold for some $n \in \mathcal{N}$. Then, there exists γ_n' for an organization $n \in \mathcal{N}$ such that for $\boldsymbol{\gamma}' = (\gamma_n', \boldsymbol{\gamma}_{-n}^{\rm NE})$,

$$V_n(f_n(\boldsymbol{\gamma}'), \widetilde{r}(\boldsymbol{\gamma}'), m_n(\boldsymbol{\gamma}', \boldsymbol{\pi}^{\rm NE})) > V_n(f_n(\boldsymbol{\gamma}^{\rm NE}), \widetilde{r}(\boldsymbol{\gamma}^{\rm NE}), m_n(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})), \quad (23)$$

which violates the assumption that $(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$ is an NE.

APPENDIX B
PROOF FOR THEOREM 1

To prove Theorem 1, we formulate an optimization problem to determine the number of training rounds r:

$$\underset{r}{\text{maximize}} \ \sum_{n\in\mathcal{N}} (U_n(r) - C_n(f_n^\circ(r), r)) \quad (24a)$$

$$\text{subject to} \ \ r \in [0, \bar{r}]. \quad (24b)$$

If r^* is an optimal solution to problem (24), then we can prove that the processing capacity vector $\boldsymbol{f}^\circ(r^*) = (f_n^\circ(r^*), n \in \mathcal{N})$ is an optimal solution to problem (8).

We now prove that for any NE of Game 1 $(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$, $\boldsymbol{f}(\boldsymbol{\gamma}^{\rm NE})$ is an optimal solution to problem (8). This is proven by showing that $r^{\rm NE} \triangleq \widetilde{r}(\boldsymbol{\gamma}^{\rm NE})$ is the optimal solution to problem (24). In the following, we first analyze problem (24) and then prove that $r^{\rm NE}$ is an optimal solution to (24).

Based on Assumption 1 and the definitions of the utility and cost functions, the objective function (24a) is concave in r. Meanwhile, since the constraint in (24b) is linear, the Karush-Kuhn-Tucker (KKT) conditions of problem (24) are sufficient for optimality. Note that the utility and cost functions are continuously differentiable. Thus, r^* is an optimal solution to problem (24) if there exist Lagrange multipliers α^* and β^* such that the following KKT conditions are satisfied:[1]

$$\sum_{n\in\mathcal{N}} \left(\frac{\partial U_n(r^*)}{\partial r^*} - \frac{\partial C_n(f_n^\circ(r^*), r^*)}{\partial r^*} \right) + \alpha^* - \beta^* = 0, \quad (25a)$$

$$r^* \geq 0, \ r^* \leq \bar{r}, \ \alpha^* \geq 0, \ \beta^* \geq 0, \quad (25b)$$

$$\alpha^* r^* = 0, \ \beta^*(r^* - \bar{r}) = 0. \quad (25c)$$

In terms of the NE $(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$ and the resulting $r^{\rm NE}$, according to Lemma 1, there exist $\boldsymbol{\alpha}^{\rm NE} = (\alpha_n^{\rm NE}, n \in \mathcal{N})$ and $\boldsymbol{\beta}^{\rm NE} = (\beta_n^{\rm NE}, n \in \mathcal{N})$ such that the following holds:

$$r^{\rm NE} = \sum_{n\in\mathcal{N}} \gamma_n^{\rm NE}/N, \quad (26a)$$

$$\frac{\partial U_n(r^{\rm NE})}{\partial r^{\rm NE}} - \frac{\partial C_n(f_n^\circ(r^{\rm NE}), r^{\rm NE})}{\partial r^{\rm NE}} + (\pi_{\mu(n+1)}^{\rm NE} - \pi_{\mu(n+2)}^{\rm NE}) + \alpha_n^{\rm NE} - \beta_n^{\rm NE} = 0, \ n \in \mathcal{N}, \quad (26b)$$

$$r^{\rm NE} \geq 0, \ r^{\rm NE} \leq \bar{r}, \alpha_n^{\rm NE} \geq 0, \ \beta_n^{\rm NE} \geq 0, \ n \in \mathcal{N}, \quad (26c)$$

$$\alpha_n^{\rm NE} r^{\rm NE} = 0, \ \beta_n^{\rm NE}(r^{\rm NE} - \bar{r}) = 0, \ n \in \mathcal{N}. \quad (26d)$$

Conditions (26b)−(26d) correspond to the KKT conditions of $\max_{r\in[0,\bar{r}]} V_n(f_n^\circ(r), r, m_n(r\mathbf{1}, \boldsymbol{\pi}^{\rm NE}))$ for each $n \in \mathcal{N}$. To prove that $r^{\rm NE}$ is an optimal solution to problem (24), let $\alpha^* = \sum_{n\in\mathcal{N}}(\pi_{\mu(n+1)}^{\rm NE} - \pi_{\mu(n+2)}^{\rm NE}) + \sum_{n\in\mathcal{N}} \alpha_n^{\rm NE} = \sum_{n\in\mathcal{N}} \alpha_n^{\rm NE}$, and let $\beta^* = \sum_{n\in\mathcal{N}} \beta_n^{\rm NE}$. Then, we have $(r^* = r^{\rm NE}, \alpha^*, \beta^*)$ satisfies the KKT conditions in (25).

APPENDIX C
PROOF FOR PROPOSITION 4

According to (14), $r^{\rm NE}$ is the optimal solution to the optimization problem $\max_{r\in[0,\bar{r}]} V_n(f_n^\circ(r), r, m_n(r\mathbf{1}, \boldsymbol{\pi}^{\rm NE}))$. Then, we can prove (16) by showing that $r(\boldsymbol{f}^{\rm NE}) = r^{\rm NE}$, and $V_n(f_n^\circ(r^{\rm NE}), r^{\rm NE}, \zeta_n^{\rm NE} r^{\rm NE}) \geq V_n(f_n', r^{\rm NE}, \zeta_n^{\rm NE} r^{\rm NE})$ for all $n \in \mathcal{N}$ under any $\boldsymbol{f}' \triangleq (f_n', n \in \mathcal{N}) \in \mathbb{R}_+^N$ that satisfies $r(\boldsymbol{f}') = r^{\rm NE}$. This inequality holds for $n \in \mathcal{N}$ because $U_n(r(\boldsymbol{f}^{\rm NE})) = U_n(r^{\rm NE}) = U_n(r(\boldsymbol{f}'))$, but $C_n(f_n^\circ(r^{\rm NE}), r(\boldsymbol{f}^{\rm NE})) \leq C_n(f_n', r(\boldsymbol{f}'))$ based on (9).

APPENDIX D
PROOF FOR LEMMA 2

The saddle point of $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda})$, denoted by $(\boldsymbol{r}^*, \boldsymbol{\lambda}^*)$, satisfies the KKT conditions of problem (17) as follows:

$$\frac{\partial U_n(r_n^*)}{\partial r_n^*} - \frac{\partial C_n(f_n^\circ(r_n^*), r_n^*)}{\partial r_n^*} - \left(\lambda_{\mu(n+2)}^* - \lambda_{\mu(n+1)}^*\right) + \widetilde{\alpha}_n^* - \widetilde{\beta}_n^* = 0, \ n \in \mathcal{N}, \quad (27a)$$

$$r_{\mu(n-2)}^* = r_{\mu(n-1)}^*, \ r_n^* \geq 0, \ r_n^* \leq \bar{r}, \ n \in \mathcal{N}, \quad (27b)$$

$$\widetilde{\alpha}_n^* \geq 0, \ \widetilde{\beta}_n^* \geq 0, \ \widetilde{\alpha}_n^* r_n^* = 0, \ \widetilde{\beta}_n^*(r_n^* - \bar{r}) = 0, \ n \in \mathcal{N}, \quad (27c)$$

where $\widetilde{\alpha}_n^*$ and $\widetilde{\beta}_n^*$ are the Lagrange multipliers corresponding to constraints $r_n \geq 0$ and $r_n \leq \bar{r}$ for $n \in \mathcal{N}$, respectively. Let $\gamma_n^{\rm NE} = r_n^*$, $\pi_n^{\rm NE} = \lambda_n^*$, $\alpha_n^{\rm NE} = \widetilde{\alpha}_n^*$, and $\beta_n^{\rm NE} = \widetilde{\beta}_n^*$ for $n \in \mathcal{N}$. Then, $(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE}, \boldsymbol{\alpha}^{\rm NE}, \boldsymbol{\beta}^{\rm NE})$ leads to (26), i.e., $(\boldsymbol{\gamma}^{\rm NE}, \boldsymbol{\pi}^{\rm NE})$ is an NE of Game 1.

APPENDIX E
PROOF FOR PROPOSITION 6

We first present an equivalent problem to problem (21). Consider a complete bipartite graph $G = (\mathcal{N}^+, \mathcal{N}^-, E)$, where $E = \mathcal{N}^+ \times \mathcal{N}^-$ denotes the set of edges in this graph. Each vertex n in sets $\mathcal{N}^+$ and $\mathcal{N}^-$ is assigned a value of $m_n^{\rm NE}$. Consider an arbitrary subset of edges $E^{\rm S} \subseteq E$ that satisfies the following condition.

Condition 1. *Let $m_{n',n}(E^{\rm S}) \in \mathbb{R}$ denote the weight assigned to edge $(n', n) \in E^{\rm S}$. Subset $E^{\rm S} \subseteq E$ must ensure that there exist weights $\boldsymbol{m}(E^{\rm S}) = (m_{n',n}(E^{\rm S}), (n', n) \in E^{\rm S})$ such that $\sum_{n'\in\mathcal{N}} m_{n',n}(E^{\rm S}) = m_n^{\rm NE}$ for all $n \in \mathcal{N}^+ \cup \mathcal{N}^-$.*

Note that $L^* \geq \max\{|\mathcal{N}^+|, |\mathcal{N}^-|\}$. If $L^* = |E^{\rm S*}| < \max\{|\mathcal{N}^+|, |\mathcal{N}^-|\}$, then there exists a vertex $n \in \mathcal{N}^+ \cup \mathcal{N}^-$

1. We use α and β to refer to the Lagrange multipliers in this appendix in order to distinguish them from the Lagrange multipliers in Section 4.

such that it is not the end point of any edge in $E^{\mathrm{S}*}$. Thus, there cannot exist any $\boldsymbol{m}(E^{\mathrm{S}})$ satisfying $m_{n',n}(E^{\mathrm{S}}) = m_n^{\mathrm{NE}}$. This contradicts to the requirement of E^{S}.

The value of L° is no larger than $|\mathcal{N}^+| + |\mathcal{N}^-|$. Specifically, with Algorithm 2, we solve the aforementioned bipartite graph problem by adding edge $(n^-(j), n^+(i))$ to E^{S} whenever $m_{n^-(j),n^+(i)}$ is set to a positive value. Note that based on Algorithm 2, when an $m_{n^-(j),n^+(i)}$ is set to a positive value (e.g., in Step 5 or 10), at least one of the corresponding remaining monetary transfer M_i^+ and M_j^- is set to be zero (e.g., in Step 6, 11, or 12). If M_i^+ is set to zero, then no other $m_{n,n^+(i)}$ for $n \in \mathcal{N}^-$ will be updated in the following iterations, i.e., no other edges $(n, n^+(i))$ for $n \in \mathcal{N}^-$ will be added to E^{S} in the following iterations. Similarly, if M_j^- is set to zero, then no other edges $(n^-(j), n)$ for $n \in \mathcal{N}^+$ will be added to E^{S} in the following iterations. Thus, at most $|\mathcal{N}^+| + |\mathcal{N}^-|$ edges are added to E^{S} before the algorithm terminates. Thus, with Algorithm 2, $|E^{\mathrm{S}}|$, or equivalently L°, is no larger than $|\mathcal{N}^+| + |\mathcal{N}^-|$. As a result, the competitive ratio is equal to $L^\circ/L^* = (|\mathcal{N}^+| + |\mathcal{N}^-|)/\max\{|\mathcal{N}^+|, |\mathcal{N}^-|\} \leq 2$.

APPENDIX F
PROOF FOR PROPOSITION 7

Algorithm 2 terminates after no more than $|\mathcal{N}^+| + |\mathcal{N}^-|$ iterations, which is on the order of $O(N)$. This is because in each iteration, at least one element of vectors $\boldsymbol{M}^+$ and $\boldsymbol{M}^-$ is set to zero. In each iteration, the existence of i and j in Step 4 can be found by first sorting the elements in vector $\boldsymbol{M}^+$ and $\boldsymbol{M}^-$. These two sorting processes have a computational complexity of $O(|\mathcal{N}^+| \log_2 |\mathcal{N}^+|)$ and $O(|\mathcal{N}^-| \log_2 |\mathcal{N}^-|)$, respectively, both of which are $O(N \log_2 N)$. After sorting, the associated M_i^+ and M_j^- (in Step 4) and the minimum elements in $\boldsymbol{M}^+$ and $\boldsymbol{M}^-$ (in Steps 9 and 10) can be found with complexity $O(N)$. Thus, the computational complexity of Algorithm 2 is $O(N(N \log_2 N + N)) = O(N^2 \log_2 N)$.