

Enabling Long-Term Cooperation in Cross-Silo Federated Learning: A Repeated Game Perspective

Ning Zhang¹, Student Member, IEEE, Qian Ma¹, Member, IEEE, and Xu Chen¹, Senior Member, IEEE

Abstract—Cross-silo federated learning (FL) is a distributed learning approach where clients of the same interest train a global model cooperatively while keeping their local data private. The success of a cross-silo FL process requires active participation of many clients. Different from cross-device FL, clients in cross-silo FL are usually organizations or companies which may execute multiple cross-silo FL processes repeatedly due to their time-varying local data sets, and aim to optimize their long-term benefits by selfishly choosing their participation levels. While there has been some work on incentivizing clients to join FL, the analysis of clients' long-term selfish participation behaviors in cross-silo FL remains largely unexplored. In this paper, we analyze the selfish participation behaviors of heterogeneous clients in cross-silo FL. Specifically, we model clients' long-term selfish participation behaviors as an infinitely repeated game, with the stage game being a selfish participation game in one cross-silo FL process (SPFL). For the stage game SPFL, we derive the unique Nash equilibrium (NE), and propose a distributed algorithm for each client to calculate its equilibrium participation strategy. We show that at the NE, clients fall into at most three categories: (i) *free riders* who do not perform local model training, (ii) a unique *partial contributor* (if exists) who performs model training with part of its local data, and (iii) *contributors* who perform model training with all their local data. The existence of free riders has a detrimental effect on achieving a good global model and sustaining other clients' long-term participation. For the long-term interactions among clients, we derive a cooperative strategy for clients which minimizes the number of free riders while increasing the amount of local data for model training. We show that enforced by a punishment strategy, such a cooperative strategy is a subgame perfect Nash equilibrium (SPNE) of the infinitely repeated game, under which some clients who are free riders at the NE of the stage game choose to be (partial) contributors. We further propose an algorithm to calculate the optimal SPNE which minimizes the number of free riders while maximizing the amount of local data for model training. Simulation results show that our derived optimal SPNE can effectively reduce the number of free riders by up to 99.3% and increase the amount of local data for model training by up to 82.3%.

Index Terms—Cross-silo federated learning, selfish participation, free rider, long-term cooperation

1 INTRODUCTION

1.1 Background and Motivations

THE rapid development of 5G and Internet of Things (IoT) technologies accelerates the generation of massive amount of user data [1], such as users' finance data in banks and patients' clinical data in hospitals. User data is of paramount importance for artificial intelligence (AI). With sufficient user data, banks can develop AI models for customized financial advice and risk control services, and hospitals can train AI-assisted diagnosis and treatment models. Traditional

machine learning approaches usually collect large amounts of raw data and train AI models on a central server, which may lead to privacy leakage. In order to protect data privacy, Google proposed federated learning (FL) [2], which is a distributed learning approach without sharing raw data.

In FL, a central server coordinates the model training of many clients [3]. In the local training steps, each client trains the model by using its local data, and then sends the updated model to the central server. In the global aggregation steps, the central server aggregates the updates and sends the aggregated global model back to clients for the next iteration. The iterations stop until a predefined stopping criterion is satisfied.

Depending on the initiator of the FL process and the type of clients, FL is classified into two types: cross-device FL and cross-silo FL [4]. In cross-device FL, as shown in Fig. 1a, the central server initiates the FL process, and clients are devices who perform local training using their local data. The central server pays each client a reward as the incentive for local training. In cross-silo FL, as shown in Fig. 1b, clients who are usually companies or organizations of the same interest, initiate the FL process and pay the central server for global aggregation.

In this paper, we study cross-silo FL, which has a wide range of practical applications. For example, WeBank and

- Ning Zhang and Qian Ma are with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510006, China. E-mail: zhangn87@mail2.sysu.edu.cn, maqian25@mail.sysu.edu.cn.
- Xu Chen is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China. E-mail: chenxu35@mail.sysu.edu.cn.

Manuscript received 29 August 2021; revised 24 December 2021; accepted 28 January 2022. Date of publication 4 February 2022; date of current version 5 June 2023.

This work was supported in part by the National Natural Science Foundation of China under Grants 62002399, U20A20159, U1711265, and 61972432, in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X355, and in part by Pearl River Talent Recruitment Program under Grant 2017GC010465.

(Corresponding author: Qian Ma.)

Digital Object Identifier no. 10.1109/TMC.2022.3148263

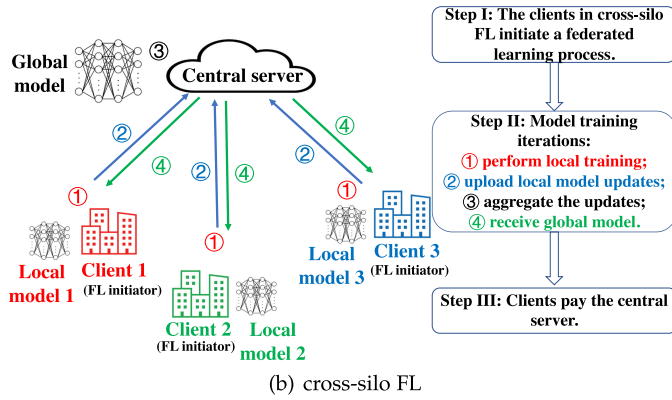
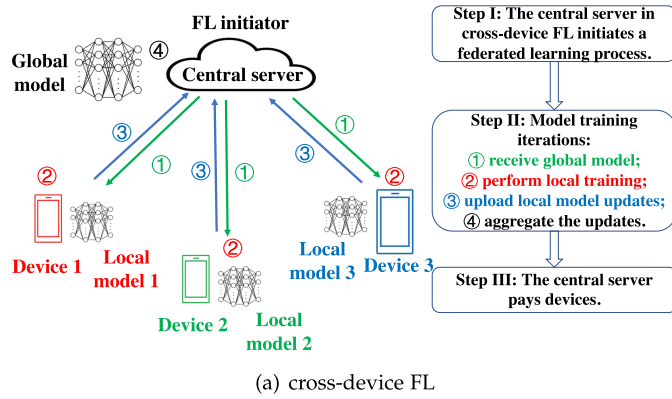
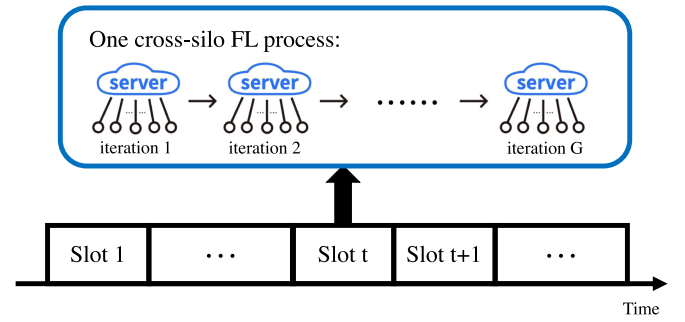


Fig. 1. An illustration of (a) cross-device FL and (b) cross-silo FL.

Swiss Re cooperatively perform data analysis for finance and insurance services [5]. NVIDIA Clara helps hospitals with different data sets train AI models by FL for mammogram assessment [6]. MELLODDY uses cross-silo FL to speed up drug research [7]. In cross-silo FL, many clients who have the same interest (e.g., training models for finance service or medical diagnosis) train a global model cooperatively, and hence usually can achieve a better model compared with the ones trained only locally [4]. Despite the popular applications of cross-silo FL seen in practice, there is little work analyzing clients' participation behaviors in cross-silo FL theoretically, which is the focus of this paper.

In cross-silo FL, since clients usually belong to different entities (e.g., companies or organizations), each client may behave selfishly to maximize its own benefit. Specifically, each client selfishly chooses its participation level (i.e., the amount of local data) for the local training steps. The total amount of local data chosen by all clients will affect the global model accuracy, while the model training leads to costs to clients. In practice, clients have heterogeneous valuations for global model accuracy and incur different computation costs [8]. Therefore, clients will make a tradeoff between the achieved global model accuracy and the incurred cost. For example, clients with high valuations for global model accuracy and low computation costs may choose a high participation level to achieve a good global model accuracy. On the other hand, clients with low valuations for global model accuracy and high computation costs may choose not to perform local training to reduce the cost, and we call such clients as *free riders*. This motivates us to study clients' selfish participation behaviors in cross-silo FL through a game-theoretic



approach. In this paper, we aim at addressing the following fundamental question in cross-silo FL.

Key Question 1. How do heterogeneous clients selfishly choose their participation levels in cross-silo FL?

Different from cross-device FL, cross-silo FL usually involves long-term repeated interactions among clients. For example, the MELLODDY project is a long-term project which involves repeated interactions among 17 partners. One reason is that clients' local data may change over time, and hence clients will perform multiple cross-silo FL processes repeatedly to adapt the global model to the time-varying local data sets. For example, hospitals constantly admit new patients and collect treatment data of these cases. Furthermore, different from cross-device FL where the central server chooses different devices to perform different FL tasks and devices aim to maximize their short-term benefits, in cross-silo FL, the same set of clients initiate FL processes repeatedly and train the global model cooperatively. Moreover, clients (e.g., organizations or companies) are usually far-sighted and aim to maximize their long-term benefits. Therefore, it is necessary to analyze clients' long-term interactions in cross-silo FL. However, the existence of free riders is detrimental to the global model for the following two reasons. First, when many clients choose to be free riders, the amount of local data for model training is small, which leads to a bad global model accuracy. Second, the behavior of free riders is unfair to the clients who use their local data to perform local trainings, which will disrupt clients' long-term cooperation. To sustain clients' long-term cooperation in cross-silo FL, we need to reduce the number of free riders. This motivates us to address the second fundamental question in cross-silo FL.

Key Question 2. How to minimize the number of free riders to sustain clients' long-term cooperation in cross-silo FL?

1.2 Contributions

In this paper, we analyze clients' participation behaviors in cross-silo FL through a game-theoretic approach. We consider an infinite time horizon which is divided into many time slots, as shown in Fig. 2. We model the interactions among clients in the infinite time horizon as a repeated game with the stage game being the selfish participation game in one cross-silo FL process (SPFL) in each time slot. Specifically, in each time slot, clients perform one cross-silo FL process which proceeds in several iterations of local training steps and global aggregation steps. Each time slot can be one month for banks [9] or one week for hospitals

[10]. Since the local data may change dynamically over time (e.g., banks update the finance data sets monthly or even weekly [9], and hospitals update the clinical data sets weekly [10]), clients need to repeatedly perform cross-silo FL processes to adapt the global model to the dynamic local data. We summarize our main contributions as follows.

- *Novel Game Analysis in Cross-Silo FL*: To the best of our knowledge, this is the first paper that provides a comprehensive game-theoretic analysis of the long-term participation behaviors of heterogeneous clients in cross-silo FL from the repeated game perspective.
- *Selfish Participation Game in One Cross-Silo FL Process (SPFL)*: For the stage game SPFL, we derive the unique Nash equilibrium (NE), and propose a distributed algorithm for each client to calculate its equilibrium participation strategy. We show that at the NE, clients fall into at most three categories: (i) clients with low valuation-computation ratios being *free riders* who do not perform local model training, (ii) a unique client (if exists) being the *partial contributor* who performs model training with part of its local data, and (iii) clients with high valuation-computation ratios being *contributors* who perform model training with all their local data.
- *Infinitely Repeated Game in the Long-Term Cross-Silo FL Processes*: In the infinitely repeated game, we derive a subgame perfect Nash equilibrium (SPNE), i.e., a cooperative strategy enforced by a punishment strategy, which achieves the minimum number of free riders while increasing the amount of local data for model training. Deriving the cooperative strategy is challenging since clients are heterogeneous and their decisions are coupled in a highly non-linear manner. We further propose an algorithm to calculate the optimal SPNE which minimizes the number of free riders while maximizing the amount of local data for model training.
- *Simulation Results*: We conduct extensive numerical evaluations and derive useful insights. First, at the NE of the stage game, the number of free riders increases with the number of clients and the size of the local data set. Our proposed optimal SPNE can effectively reduce the number of free riders by up to 99.3%, and increase the amount of local data for model training by up to 82.3%. The optimal SPNE performs well even when the number of clients and the size of local data set are large, which shows its scalability. Second, compared with other methods, our proposed optimal SPNE has distinct advantages in the number of contributors and the total amount of training data.

1.3 Related Work

FL has drawn researchers' attention in recent years. McMahhan *et al.* propose and illustrate the effectiveness of FL [11]. In the following, we discuss related work regarding FL optimization, incentive mechanism design, the free rider issue, and cross-silo FL.

FL Optimization. Some papers aim to optimize different objectives in FL, such as the learning time, the model accuracy, and the energy consumption of local devices. Zhong *et al.* propose the *P-Fedavg* algorithm for the Parallel FL system with provable convergence rate [12]. Wang *et al.* propose a GCN-based algorithm to derive the optimal device sampling strategy considering D2D offloading to maximize the training accuracy [13]. Mo *et al.* optimize the energy efficiency by balancing the energy tradeoff between communication and computation in FL [14]. Luo *et al.* propose a sampling based algorithm to minimize the total cost with marginal overhead [15].

Incentive Mechanism Design. There are many papers focusing on the incentive mechanism design for FL. For example, Ding *et al.* in [16] consider contract design for the central server in cross-device FL under multi-dimensional private information. Sun *et al.* in [17] propose a contract-based incentive mechanism to customize the payment for each participating worker considering personalized privacy preference. Zhan *et al.* in [18] propose a DRL-based incentive mechanism to capture the tradeoff between payment and training time in FL without prior knowledge of edge nodes. Zeng *et al.* in [19] propose an auction-based lightweight incentive mechanism considering multi-dimensional and dynamic edge resources to guide the server to effectively choose clients to participate in FL. Deng *et al.* in [20] propose an auction-based quality-aware system called FAIR to estimate clients' learning qualities and encourage high-quality clients' participation. Zhan *et al.* in [21] summarize the papers on incentive mechanism design for FL from three aspects: clients' data contribution, clients' reputation, and resource allocation. However, none of these papers solves the free-rider problem in federated learning. Our work complements the research in this area.

Free Rider Issue. Free riders are those who benefit from resources or public goods but refuse to pay for them [22], which have been extensively studied in peer-to-peer systems [23]. Several papers study how to identify free riders in FL. Lin *et al.* reveal the free-riding attack in FL and propose a novel anomaly detection technique using autoencoders to identify free riders [24]. Huang *et al.* design a protection mechanism named Gradient Auditing to detect and punish the free-riding behavior [25]. Fraboni *et al.* propose to establish a routine practice to inspect clients' distribution for the detection of free-rider attacks in FL [26]. Although these papers notice the existence of free riders in FL, they only focus on identifying free riders rather than addressing the problem effectively.

Cross-Silo FL. There have been some researches analyzing cross-silo FL. Chen *et al.* propose FOCUS for cross-silo FL to solve the problem of noisy local labels [27]. Heikkilä *et al.* combine additively homomorphic secure summation protocols with differential privacy in cross-silo FL to learn complex models while guaranteeing privacy of local data [28]. Majeed *et al.* build a cross-silo horizontal federated model for traffic classification [29]. Zhang *et al.* present a system solution BatchCrypt for cross-silo FL to reduce the encryption and communication overhead caused by additively homomorphic encryption [30]. Tang *et al.* propose an incentive mechanism considering the public goods feature of cross-silo FL [8]. However, none of the above papers

analyze the long-term selfish participation behaviors of clients in cross-silo FL.

In summary, existing papers mainly focus on cross-device FL, and none of these papers studies clients' long-term selfish participation behaviors in cross-silo FL. As far as we know, this is the first work that analyzes the long-term participation behaviors of heterogeneous clients in cross-silo FL and proposes a strategy to reduce the number of free riders through the game-theoretic approach.

We organize our paper as follows. In Section 2, we present the system model. Then we analyze the stage game SPFL in Section 3. We analyze the infinitely repeated game in Section 4. We present simulation results in Section 5 and conclude in Section 6. Due to space constraints, we relegate all the proofs to the supplementary material, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TMC.2022.3148263>.

2 SYSTEM MODEL

We consider a set $\mathcal{N} = \{1, 2, \dots, N\}$ of clients (e.g., companies or organizations) with the same interest participating in the cross-silo FL processes. Each client has some local data which may change over time, and clients will perform multiple cross-silo FL processes repeatedly to adapt the global model to the time-varying local data sets. Since clients are far-sighted and aim to optimize their long-term benefits, we study clients' long-term interactions in cross-silo FL processes in the infinite time horizon which is divided into time slots. In the following, we first describe a cross-silo FL process in one time slot, then model the cost of clients in a cross-silo FL process, and finally show the interactions among clients in two time scales.

2.1 A Cross-Silo FL Process in One Time Slot

As shown in Fig. 2, each time slot corresponds to one cross-silo FL process which proceeds in training iterations. We first introduce the objective of cross-silo FL, and then describe the iteration process.

In a cross-silo FL process, clients of the same interest cooperatively train a global model represented by a parameter vector w . Each client $n \in \mathcal{N}$ has a local data set \mathcal{D}_n , where the number of local data samples is $D_n \triangleq |\mathcal{D}_n|$.¹ Each client can choose a subset $\mathcal{X}_n \subseteq \mathcal{D}_n$ of local data for model training. Let x_n denote the size of the chosen subset of local data, i.e., $x_n \triangleq |\mathcal{X}_n|$, and let d_{ni} denote the i th data sample in set \mathcal{X}_n . Cross-silo FL aims to find the optimal global model w^* that minimizes the global loss function [4]:

$$L(w) = \sum_{n \in \mathcal{N}} \frac{x_n}{\sum_{n' \in \mathcal{N}} x_{n'}} L_n(w). \quad (1)$$

Here $L_n(w)$ is the local loss function of client n

$$L_n(w) = \frac{1}{x_n} \sum_{d_{ni} \in \mathcal{X}_n} l(w; d_{ni}). \quad (2)$$

Here $l(w; d_{ni})$ is the loss function for data sample d_{ni} under w .

1. Although different clients have different local data, we assume that the data is i.i.d. across all clients [31].

To achieve the optimal global model w^* , cross-silo FL proceeds in training iterations. One widely adopted algorithm to derive w^* is the FedAvg algorithm [11].² In each training iteration r , clients perform local model trainings over the previous global model w^{r-1} with the chosen subsets of local data by using the mini-batch stochastic gradient descent (SGD) method [32]. Clients derive the updated local models $w_n^r, \forall n \in \mathcal{N}$, and send them³ to the central server for global aggregation. The central server derives the updated global model $w^r = \sum_{n \in \mathcal{N}} \frac{x_n}{\sum_{n' \in \mathcal{N}} x_{n'}} w_n^r$ [11]. The above cross-silo FL process causes costs to clients, which we will introduce in detail next.

2.2 Cost of Clients in a Cross-Silo FL Process

In cross-silo FL, clients incur the following costs: the model accuracy loss, the computation cost, the communication cost, and the payment to the central server.

2.2.1 Model Accuracy Loss

The purpose of clients to perform cross-silo FL is to achieve a global model with good accuracy, i.e., a global model with a small accuracy loss [31]. The global model accuracy loss can be calculated as $L(w^G) - L(w^*)$, where $L(w^G)$ and $L(w^*)$ are the global losses under parameters w^G and w^* , respectively, and G is the number of training iterations in a cross-silo FL process. According to [34], the expected global model accuracy loss is bounded by $O(1/\sqrt{BG} + 1/G)$, where B is the total batch size that clients use for model trainings, i.e., $B = \sum_{n \in \mathcal{N}} x_n$.⁴ Hence, the expected global model accuracy loss decreases with the amount of local data $\sum_{n \in \mathcal{N}} x_n$ for model trainings and the number of training iterations G . In this paper, we denote the model accuracy loss of each client $n \in \mathcal{N}$ as $A(x_n, x_{-n})$, which depends on all clients' chosen amounts of local data for model trainings and can be calculated as follows:⁵

$$A(x_n, x_{-n}) = \frac{1}{\sqrt{(x_n + \sum_{n' \in \mathcal{N}, n' \neq n} x_{n'})G}} + \frac{1}{G}, \quad (3)$$

where $x_{-n} = \{x_1, x_2, \dots, x_{n-1}, x_{n+1}, \dots, x_N\}$.

2.2.2 Computation Cost

The local model training steps consume local computation resources. In a cross-silo FL process, the CPU energy consumption of each client $n \in \mathcal{N}$, which depends on its chosen

2. In cross-silo FL, we assume that all clients participate in training iterations, and clients perform synchronous update scheme [4].

3. We assume that clients will truthfully report their local models to the central server and the central server can verify clients' contributions (e.g., by the Trusted Execution Environments proposed in [33]).

4. Note that in FL, the central server knows clients' chosen amounts of local data for model trainings, i.e., the values of $x_n, \forall n \in \mathcal{N}$, which are the weights in the global aggregation steps [8], [31].

5. In this paper, we assume a complete information setting where each client $n \in \mathcal{N}$ knows other clients' chosen amounts of local data, i.e., the values of $x_{n'}, \forall n' \in \mathcal{N}, n' \neq n$. This information can be announced by the central server to clients. Note that our proposed algorithms later in the paper which calculate the equilibrium strategy for each client do not require such information. The analysis in the complete information setting provides useful insights for practical cross-silo FL systems. For the analysis of the incomplete information setting, we will leave it as future work.

amount of local data x_n for model training, can be calculated as [35]

$$\mathcal{E}(x_n) = \frac{\varsigma_n}{2} \mu_n \vartheta_n^2 x_n, \quad (4)$$

where ς_n is a coefficient depending on the client's computing chip architecture, μ_n is the number of CPU cycles of client n to perform the model training on one local data sample, and ϑ_n is the CPU processing speed (in cycles per second) of client n . We denote the *computation cost coefficient* of client n as $E_n = \frac{\varsigma_n}{2} \mu_n \vartheta_n^2$, and the computation cost of client n can be calculated as $\mathcal{E}(x_n) = E_n x_n$.

2.2.3 Communication Cost

In a cross-silo FL process, clients send their local models $w_n^r, \forall n \in \mathcal{N}$, to the central server for aggregation, and receive the updated global model w^r for the next training iteration, for all $r = 1, 2, \dots, G$. Since clients in cross-silo FL are companies or organizations, the data transmission between clients and the central server can be through either wired networks (e.g., through the high-speed wired connections [36]) or wireless networks (e.g., using transmission protocols TDMA, OFDMA or NOMA [37]). When transmitting model updates, clients usually have different data transmission rates which depend on their transmission power and channel coefficients in wireless networks [38]. In this case, clients will incur heterogeneous communication costs (e.g., different transmission delays). We denote the communication cost of each client $n \in \mathcal{N}$ as C_n . Since clients' model updates have the same size, C_n does not depend on the participation strategy x_n and is a constant in our paper.

2.2.4 Payment to the Central Server

Clients need to pay the central server for global model aggregation. We assume that the payment of each client is p .

In summary, we define the total cost of each client $n \in \mathcal{N}$ as follows:

$$F_n(x_n, \mathbf{x}_{-n}) = \rho_n A(x_n, \mathbf{x}_{-n}) + \mathcal{E}(x_n) + C_n + p. \quad (5)$$

Here ρ_n denotes client n 's valuation for the model accuracy, which describes how important the model accuracy is to the client [8]. For example, ρ_n can be a bank's customer churn per unit of model accuracy loss when using the risk control service trained by cross-silo FL, or a pharmaceutical company's unit revenue loss when using the drug research model trained by cross-silo FL.

2.3 Interactions Among Clients in Two Time Scales

In this subsection, we consider the interactions among clients in two time scales as shown in Fig. 2. Specifically, we describe the behaviors of clients in each time slot and in the infinite time horizon, respectively.

We first analyze the interactions among clients in one time slot. As in [9], [10], the length of a time slot can be a month for banks or a week for hospitals. One time slot corresponds to one cross-silo FL. At the beginning of the time slot, each client n chooses the amount of local data x_n for model training (i.e., its participation level in the cross-silo FL process), to minimize its total cost calculated in (5), considering the participation behaviors of other clients. When

clients are myopic and only care about their costs in the current time slot, we model their participation behaviors as a selfish participation game in one cross-silo FL process (SPFL), to be introduced in detail in Section 3.

We then analyze the interactions among clients in the infinite time horizon which is divided into many time slots. In the infinite time horizon, each client's local data set changes over time, i.e., $\mathcal{D}_n^{t-1} \neq \mathcal{D}_n^t, \forall n \in \mathcal{N}$, where \mathcal{D}_n^{t-1} and \mathcal{D}_n^t are the local data sets of client n in time slot $t-1$ and time slot t , respectively. We assume that the number of local data samples remains unchanged, i.e., $|\mathcal{D}_n^{t-1}| = |\mathcal{D}_n^t|, \forall n \in \mathcal{N}$.⁶ To adapt the global model to the time-varying data sets, clients perform cross-silo FL processes repeatedly. Specifically, at the beginning of each time slot t , clients receive the global model w^{t-1} in the previous time slot, and then perform a cross-silo FL process using their local data (as described in the selfish participation game SPFL) to derive the global model w^t . In the infinite time horizon, clients are far-sighted, and each client n chooses the amount of local data for model training to minimize its long-term discounted total cost. We model clients' long-term selfish participation behaviors as an infinitely repeated game with the stage game being SPFL, which will be introduced in detail in Section 4.

3 STAGE GAME ANALYSIS

In this section, we analyze the stage game, i.e., the selfish participation game in one cross-silo FL process (SPFL). We derive the unique Nash equilibrium of the stage game SPFL, and design a distributed algorithm for each client to compute its equilibrium participation strategy.

3.1 Stage Game Modeling

In a cross-silo FL process, clients selfishly choose their participation levels to minimize their own costs. We model the behaviors of clients in each time slot as a selfish participation game as follows:

Game 1 (Selfish Participation Game in a Cross-Silo FL Process (SPFL)).

- *Players:* the set \mathcal{N} of clients.
- *Strategies:* each client $n \in \mathcal{N}$ chooses the amount of local data $x_n \in [0, D_n]$ for model training.
- *Objectives:* each client $n \in \mathcal{N}$ aims to minimize its total cost $F_n(x_n, \mathbf{x}_{-n})$ defined in (5).

Here D_n is the total number of data samples in client n 's local data set \mathcal{D}_n , i.e., $D_n = |\mathcal{D}_n|$.

3.2 Nash Equilibrium of SPFL

Next we define the Nash equilibrium (NE) of SPFL and derive the unique NE later.

Definition 1 (Nash Equilibrium).

A Nash equilibrium of Game 1 is a strategy profile $\mathbf{x}^* = \{x_n^* : \forall n \in \mathcal{N}\}$ such that for each client $n \in \mathcal{N}$

$$F_n(x_n^*, \mathbf{x}_{-n}^*) \leq F_n(x_n, \mathbf{x}_{-n}^*), \text{ for all } x_n \in [0, D_n].$$

6. For banks or hospitals, the number of users does not change much from month to month, so we assume that the number of local data samples remains the same.

At the NE, each client's strategy is the best response to other clients' strategies. In Lemma 1, we characterize the best response of each client $n \in \mathcal{N}$ that minimizes its total cost given all other clients' strategies \mathbf{x}_{-n} .

Lemma 1.

The best response of each client $n \in \mathcal{N}$ in Game 1 is

$$x_n^{BR}(\mathbf{x}_{-n}) = \min \left\{ D_n, \max \left\{ 3 \sqrt{\frac{\rho_n^2}{4GE_n^2}} - \sum_{n' \in \mathcal{N}, n' \neq n} x_{n'}, 0 \right\} \right\}. \quad (6)$$

Proof. See Appendix A in the supplementary material, available online. \square

Client n 's best response $x_n^{BR}(\mathbf{x}_{-n})$ increases with its valuation ρ_n for model accuracy, while decreases with its computation cost coefficient E_n , the number of iterations G , and the total amount of local data $\sum_{n' \in \mathcal{N}, n' \neq n} x_{n'}$ that other clients choose for model training. Intuitively, when client n has a high valuation for the global model accuracy, client n may choose a large amount of local data for model training to achieve a good global model. On the contrary, when client n has a large computation cost coefficient E_n , client n may choose a small amount of local data for training to reduce the computation cost. Furthermore, when the number of iterations G is large, clients can achieve a good global model with a small amount of local data as shown in (3). Similarly, when the total amount of local data from other clients is large, the accuracy loss in (3) is small and client n will choose a small amount of local data for model training to reduce the computation cost.

Next we derive the NE of Game 1. For simplicity of analysis, we assume that $D_n = D, \forall n \in \mathcal{N}$. For each client $n \in \mathcal{N}$, we define $h_n \triangleq 3 \sqrt{\frac{\rho_n^2}{4GE_n^2}}$, which depends on $\frac{\rho_n}{E_n}$, the valuation-computation ratio (i.e., the ratio of the valuation parameter to the computation cost coefficient). Without loss of generality, we assume that $\frac{\rho_1}{E_1} \leq \frac{\rho_2}{E_2} \leq \dots \leq \frac{\rho_N}{E_N}$.

Theorem 1. A Nash equilibrium \mathbf{x}^* always exists in Game 1, and falls into one of the following two cases.

- Case I: If there exists a critical client $k \in \mathcal{N}$ that satisfies $(N-k)D \leq h_k \leq (N+1-k)D$, then the NE $\mathbf{x}^* = \{x_n^* : \forall n \in \mathcal{N}\}$ is

$$x_n^* = \begin{cases} 0, & \text{if } n < k; \\ h_k - (N-k)D, & \text{if } n = k; \\ D, & \text{if } n > k. \end{cases} \quad (7)$$

- Case II: If the critical client k in Case I does not exist, then there must exist a client $m \in \mathcal{N}$ that satisfies $h_m < (N-m)D < h_{m+1}$, and in this case, the NE $\mathbf{x}^* = \{x_n^* : \forall n \in \mathcal{N}\}$ is

$$x_n^* = \begin{cases} 0, & \text{if } n \leq m; \\ D, & \text{if } n > m. \end{cases} \quad (8)$$

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

Theorem 1 shows that the NE of Game 1 falls into two cases depending on whether a critical client k exists or not. If such a critical client k exists, clients can be divided into three categories at equilibrium: (i) clients whose valuation-computation ratios are lower than that of the critical client k choose to be *free riders* that do not perform local model training, (ii) the unique critical client k chooses to be a *partial contributor* who performs model training with part of its local data, and (iii) clients whose valuation-computation ratios are higher than that of the critical client k choose to be *contributors* who perform model training with all their local data. If the critical client k does not exist, clients can be divided into two categories at equilibrium: (i) clients with low valuation-computation ratios choose to be free riders, and (ii) clients with high valuation-computation ratios choose to be contributors. Note that at equilibrium of Game 1, it is not possible that all clients choose to be free riders, as shown in the following corollary.

Corollary 1. At equilibrium of Game 1, the client with the highest valuation-computation ratio chooses a positive amount of local data for model training, i.e., $x_N^* > 0$.

The significance of the above result is to establish that at equilibrium of Game 1, there will always be a positive amount of local data chosen by clients to perform model training in cross-silo FL. Specifically, the client N who has the highest valuation-computation ratio will never choose to be a free rider. Even if all other clients choose to be free riders at equilibrium, client N will choose a positive amount of local data for model training, otherwise its model accuracy loss in (3) and its total cost in (5) will go to infinity.

At equilibrium, however, clients with low valuation-computation ratios may choose to be free riders. This is a result of the tradeoff between the model accuracy loss and the computation cost. Specifically, given the global model aggregated from the local models of the (partial) contributors, clients with low valuation-computation ratios choose not to perform local trainings to avoid the computation costs. The behaviors of free riders only minimize their own costs, but cannot improve the global model accuracy. As discussed in Section 1.1, the existence of free riders hampers clients' long-term participation in cross-silo FL. Later in Section 4, we will analyze the interactions among clients in the infinite time horizon, and design a cooperative strategy to reduce the number of free riders.

Next we discuss the uniqueness of the NE of Game 1.

Theorem 2. Game 1 admits a unique Nash equilibrium if $\frac{\rho_1}{E_1} < \frac{\rho_2}{E_2} < \dots < \frac{\rho_N}{E_N}$.

Proof. See Appendix C in the supplementary material, available online. \square

Here $\frac{\rho_1}{E_1} < \frac{\rho_2}{E_2} < \dots < \frac{\rho_N}{E_N}$ is a mild constraint which can be satisfied in most cases in practice.⁷ To reach the NE of

7. The constraint that $\{\frac{\rho_n}{E_n} : \forall n \in \mathcal{N}\}$ can be ranked in a strictly ascending order is mild, because clients usually have different valuations ρ_n for global model accuracy as well as heterogeneous computation cost coefficients E_n . When the constraint $\frac{\rho_1}{E_1} < \frac{\rho_2}{E_2} < \dots < \frac{\rho_N}{E_N}$ is not satisfied, the Nash equilibrium may not be unique. Consider an example with $N = 2$ clients, and $\frac{\rho_1}{E_1} = \frac{\rho_2}{E_2}$, $h_1 = h_2 = 10$, $D_1 = D_2 = 10$. In this example, $(x_1^*, x_2^*) = (4, 6)$ is a Nash equilibrium of Game 1, and $(x_1^*, x_2^*) = (5, 5)$ is also a Nash equilibrium of Game 1.

Game 1, we prove in Appendix D, available in the online supplemental material, that the stage game SPFL is a concave game and the best response update process converges to the NE of Game 1. Next we will design a distributed algorithm to calculate the equilibrium participation strategy for each client.

Algorithm 1. A Distributed Algorithm to Compute Equilibrium Strategy for Each Client in Game 1

Input: $\rho_n, E_n, \forall n \in \mathcal{N}, G, N, D$
Output: $x_n^*, \forall n \in \mathcal{N}$

- 1: Each client $n \in \mathcal{N}$ reports the value of $\frac{\rho_n}{E_n}$ to the central server.
- 2: The central server sorts the clients in an ascending order of $\frac{\rho_n}{E_n}, \forall n \in \mathcal{N}$, and sends each client its index n as well as the total number of clients N .
- 3: **for** $n = 1 : N$ **do**
- 4: Calculate $h_n = \sqrt[3]{\frac{\rho_n^2}{4GE_n^2}}$,
- 5: **if** $h_n < (N - n)D$ **then**
- 6: $x_n^* = 0$;
- 7: **else if** $h_n > (N - n + 1)D$ **then**
- 8: $x_n^* = D$;
- 9: **else**
- 10: $x_n^* = h_n - (N - n)D$.
- 11: **end**
- 12: **end**

3.3 A Distributed Algorithm to Compute Equilibrium Strategy

In this section, we design a distributed algorithm (in Algorithm 1) for clients to compute the equilibrium strategy of Game 1. In the distributed algorithm, each client calculates its equilibrium participation strategy based on its own information, without knowing other clients' participation decisions or parameter information (e.g., the valuation parameters for global model accuracy and the computation cost coefficients).

To calculate the equilibrium strategy, each client $n \in \mathcal{N}$ needs to know the total number of clients N and the ranking of its valuation-computation ratio $\frac{\rho_n}{E_n}$ in all clients (i.e., the index n), which can be obtained by reporting its ratio $\frac{\rho_n}{E_n}$ to the central server for ordering⁸ (Lines 1-2 in Algorithm 1). The equilibrium strategy of each client $n \in \mathcal{N}$ depends on the value of h_n . If $h_n < (N - n)D$, client n chooses to be a free rider, i.e., $x_n^* = 0$ (Lines 5-6 in Algorithm 1). If $h_n > (N - n + 1)D$, client n chooses to be a contributor, i.e., $x_n^* = D$ (Lines 7-8 in Algorithm 1). Otherwise, client n chooses to be a partial contributor, i.e., $x_n^* = h_n - (N - n)D$ (Lines 9-10 in Algorithm 1). Note that clients do not need to know whether the critical client k exists at equilibrium, and Algorithm 1 covers both cases of the equilibrium described in Theorem 1.

Algorithm 1 has a linear complexity of $\mathcal{O}(N)$. Specifically, computing the value of h_n and the value of x_n^* for each client $n \in \mathcal{N}$ in Lines 4-11 has a complexity of $\mathcal{O}(1)$. With

the for loop operation in Line 3, the overall complexity of Algorithm 1 is $\mathcal{O}(N)$. Thus it is scalable with the number of clients in cross-silo FL.

As discussed in Section 2.3, the interactions among clients in the infinite time horizon are inherently a repeated process. Since clients' local data may change over time, they need to adapt the global model to the local data sets constantly. In the next section, we analyze clients' long-term selfish participation behaviors.

4 REPEATED GAME ANALYSIS

In this section, we analyze the long-term interactions among clients in cross-silo FL processes. We model clients' selfish participation behaviors as a repeated game in the infinite time horizon, with the stage game being the SPFL (i.e., Game 1). We first model the repeated game, and then derive a cooperative participation strategy that can minimize the number of free riders while increasing the amount of local data for model training. We derive a subgame perfect Nash equilibrium (SPNE) of the repeated game which enforces the cooperative participation strategy by a punishment strategy. Finally, we propose an algorithm to calculate the optimal SPNE that can minimize the number of free riders while maximizing the amount of local data for model training.

4.1 Repeated Game Modeling

As shown in Fig. 2, we consider clients' interactions in the infinite time horizon which consists of infinitely many time slots. Each time slot corresponds to one cross-silo FL process. In the infinite time horizon, each client chooses its participation level (i.e., the amount of local data for model training) for each cross-silo FL process to minimize its long-term discounted total cost.

We first calculate the long-term discounted total cost for each client. We denote the participation strategy profile of all clients in time slot t as $\mathbf{x}^t = \{x_n^t : \forall n \in \mathcal{N}\}$. For each client $n \in \mathcal{N}$, its local data set changes over time while keeping the size unchanged, i.e., $\mathcal{D}_n^{t-1} \neq \mathcal{D}_n^t, |\mathcal{D}_n^{t-1}| = |\mathcal{D}_n^t| = D_n$. Therefore, we have $x_n^t \in [0, D_n]$ for each time slot t . We denote the strategy profile history up to time slot t as

$$\mathbf{s}^t \triangleq \{\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^{t-1}\}.$$

Then for each client $n \in \mathcal{N}$, its long-term discounted total cost is

$$C_n(\mathbf{s}^\infty) = \sum_{t=0}^{\infty} \delta_n^t F_n(\mathbf{x}^t), \quad (9)$$

where $\delta_n \in [0, 1)$ is the discount factor [40] of client n , and $F_n(\mathbf{x}^t)$ is the cost of client n in time slot t calculated in (5).

We model the interactions among clients in the infinite time horizon as a repeated game as follows.

Game 2 (Repeated Game in Infinitely Many Cross-Silo FL Processes).

- *Players:* the set \mathcal{N} of clients.
- *Strategies:* each client $n \in \mathcal{N}$ chooses the amount of local data $x_n^t \in [0, D_n]$ for model training in each time slot t .

8. We assume that each client $n \in \mathcal{N}$ will truthfully report its ratio $\frac{\rho_n}{E_n}$ to the central server. How to incentivize clients to truthfully reveal their information [39] is beyond the scope of this paper.

- *Histories*: the strategy profile history s^t till time slot t .
- *Objectives*: each client $n \in \mathcal{N}$ aims to minimize its long-term discounted total cost $C_n(s^\infty)$ defined in (9).

In the repeated game (Game 2), clients train the global model repeatedly to adapt the model to clients' time-varying local data sets by participating in infinitely many cross-silo FL processes. We aim to find the subgame perfect Nash equilibrium (SPNE) of Game 2. According to Folk Theorem [40], under proper discount factors $\delta_n, \forall n \in \mathcal{N}$, any feasible and individually rational participation strategy profile can be the SPNE of the infinitely repeated game [41]. Since the success of cross-silo FL depends on the active long-term participation of clients, we will characterize the SPNE that can minimize the number of free riders while increasing the chosen amount of local data for model training. Next in Section 4.2, we derive a cooperative participation strategy in each cross-silo FL process, i.e., a strategy profile where clients cooperate to minimize the number of free riders. Later in Section 4.3, we show that the cooperative participation strategy can be enforced as the SPNE of Game 2 by a punishment strategy.

4.2 Cooperative Participation Strategy

Now we derive the cooperative participation strategy $x^{\text{coop},t}$ in each time slot t . The cooperative participation strategy aims to minimize the number of free riders while reducing the cost of each client compared with the cost at the NE of the stage game SPFL, i.e., $F_n(x^{\text{coop},t}) < F_n(x^{*,t})$ for all $n \in \mathcal{N}$. Here $x^{*,t}$ is the NE of the stage game SPFL in time slot t . In this subsection, we focus on a time slot t . When there is no confusion, we ignore the time index and write $x^{\text{coop},t}$ and $x^{*,t}$ as x^{coop} and x^* , respectively.

We first analyze the cooperative participation strategy for the case where the critical client k exists at the NE of Game 1 (as shown in Case I in Theorem 1). For each client $n \in \mathcal{N}$, we

define $B_n \triangleq \frac{2\sqrt{G((N-k)D+x_k^*)^3}}{k-n}$ and $O_n \triangleq \frac{D\sqrt{G}}{\frac{1}{\sqrt{(N-k)D+x_k^*}} - \frac{1}{\sqrt{(N-n)D+x_k^*}}}$,

where x_k^* is the participation strategy of the critical client k at the NE of Game 1. In this case, we derive the cooperative strategy as follows.

Theorem 3. *When the critical client k exists at the NE of Game 1 (as shown in Case I in Theorem 1), we find the client l which satisfies*

$$l = \min \left\{ n \in \mathcal{N} : n < k, \text{ and } \frac{\rho_n}{E_n} > B_n \right\}. \quad (10)$$

Then the cooperative participation strategy $x^{\text{coop}} = \{x_n^{\text{coop}} : \forall n \in \mathcal{N}\}$ that minimizes the number of free riders is

$$x_n^{\text{coop}} = \begin{cases} 0, & \text{if } n < l; \\ x^{\text{coop}}, & \text{if } l \leq n < k; \\ x_k^*, & \text{if } n = k; \\ D, & \text{if } n > k. \end{cases} \quad (11)$$

Here x^{coop} is a cooperative participation level that satisfies $x^{\text{coop}} \leq x_l^{\text{th}}$, and x_l^{th} is the maximum amount of local data that client l can choose under the cooperative strategy, calculated as follows:

$$x_l^{\text{th}} = \begin{cases} x_l^{\text{th}} \left(\frac{\rho_l}{E_l} \right), & \text{if } B_l < \frac{\rho_l}{E_l} \leq O_l; \\ D, & \text{if } \frac{\rho_l}{E_l} > O_l. \end{cases} \quad (12)$$

Here $x_l^{\text{th}} \left(\frac{\rho_l}{E_l} \right)$, which depends on the valuation-computation ratio $\frac{\rho_l}{E_l}$, is the unique non-zero solution to the following implicit equation:

$$\frac{\rho_l}{E_l} = \frac{\sqrt{G}x_l^{\text{th}}}{\frac{1}{\sqrt{(N-k)D+x_k^*}} - \frac{1}{\sqrt{(N-k)D+x_k^*+(k-l)x_l^{\text{th}}}}}. \quad (13)$$

Proof. See Appendix E in the supplementary material, available online. \square

Theorem 3 shows that when the critical client k exists at the NE of Game 1, we can derive a cooperative participation strategy under which clients fall into at most four categories: (i) the free riders at the NE of Game 1 who satisfy $\frac{\rho_n}{E_n} \leq B_n$ still choose to be *free riders* under the cooperative strategy, (ii) the free riders at the NE of Game 1 who satisfy $\frac{\rho_n}{E_n} > B_n$ choose to be *converted contributors* who perform model training with a positive amount⁹ of local data x^{coop} , (iii) the unique critical client k chooses to be a *partial contributor* who performs model training with the amount of local data x_k^* , and (iv) the contributors at the NE of Game 1 still choose to be *contributors* who perform model training with all their local data.

Compared with the NE of Game 1 in Theorem 1, Theorem 3 shows that under the cooperative participation strategy, we have the following improved results. First, the number of free riders is reduced from $k-1$ to $l-1$ where $l \leq k$. Second, the amount of local data chosen by the converted contributors increases from 0 to $x^{\text{coop}} > 0$. The following corollary shows that under certain conditions, the cooperative strategy can reduce the number of free riders to 0.

Corollary 2. *Under the cooperative strategy in Theorem 3, no client chooses to be a free rider if*

$$\frac{\rho_1}{E_1} > B_1 \triangleq \frac{2\sqrt{G((N-k)D+x_k^*)^3}}{k-1}.$$

Furthermore, all clients choose to be contributors, i.e., $x_n^{\text{coop}} = D, \forall n \in \mathcal{N}$, if

$$\frac{\rho_1}{E_1} > O_1 \triangleq \frac{D\sqrt{G}}{\frac{1}{\sqrt{(N-k)D+x_k^*}} - \frac{1}{\sqrt{(N-1)D+x_k^*}}}.$$

The above result shows that when client 1's ratio $\frac{\rho_1}{E_1}$ is high, all clients choose to be (converted/partial) contributors under the cooperative participation strategy, and there is no free rider in cross-silo FL.

Next, we analyze the cooperative participation strategy for the case where the critical client k does not exist at the NE of Game 1 (as shown in Case II in Theorem 1). For each client $n \in \mathcal{N}$, we define $B'_n \triangleq \frac{2\sqrt{G(N-m)^3D^3}}{m-n+1}$ and $O'_n \triangleq \frac{D\sqrt{G}}{\frac{1}{\sqrt{(N-m)D}} - \frac{1}{\sqrt{(N-n+1)D}}}$.

Here m is the index of the client who satisfies $h_m <$

9. We assume that for fairness, converted contributors choose the same amount of local data for model training. Fairness is an important problem in FL [42], which is beyond the scope of this paper.

$(N - m)D < h_{m+1}$. In this case, we derive the cooperative strategy as follows.

Theorem 4. When there is a client m that satisfies $h_m < (N - m)D < h_{m+1}$ at the NE of Game 1 (as shown in Case II in Theorem 1), we find the client l which satisfies

$$l = \min \left\{ n \in \mathcal{N} : n \leq m, \text{ and } \frac{\rho_n}{E_n} > B'_n \right\}. \quad (14)$$

Then the cooperative participation strategy $\mathbf{x}^{coop} = \{x_n^{coop} : \forall n \in \mathcal{N}\}$ that minimizes the number of free riders is

$$x_n^{coop} = \begin{cases} 0, & \text{if } n < l; \\ x^{coop}, & \text{if } l \leq n \leq m; \\ D, & \text{if } n > m. \end{cases} \quad (15)$$

Here x^{coop} is a cooperative participation level that satisfies $x^{coop} \leq x_l^{th}$, and x_l^{th} is the maximum amount of local data that client l can choose under the cooperative strategy, calculated as follows:

$$x_l^{th} = \begin{cases} x_l^{th} \left(\frac{\rho_l}{E_l} \right), & \text{if } B'_l < \frac{\rho_l}{E_l} \leq O'_l; \\ D, & \text{if } \frac{\rho_l}{E_l} > O'_l. \end{cases} \quad (16)$$

Here $x_l^{th} \left(\frac{\rho_l}{E_l} \right)$, which depends on the valuation-computation ratio $\frac{\rho_l}{E_l}$, is the unique non-zero solution to the following implicit equation:

$$\frac{\rho_l}{E_l} = \frac{\sqrt{G}x_l^{th}}{\frac{1}{\sqrt{(N-m)D}} - \frac{1}{\sqrt{(N-m)D + (m-l+1)x_l^{th}}}}. \quad (17)$$

Proof. See Appendix F in the supplementary material, available online. \square

Theorem 4 shows that when the critical client k does not exist at the NE of Game 1, we can derive a cooperative participation strategy under which clients fall into at most three categories: (i) the free riders at the NE of Game 1 who satisfy $\frac{\rho_n}{E_n} \leq B'_n$ still choose to be *free riders* under the cooperative strategy, (ii) the free riders at the NE of Game 1 who satisfy $\frac{\rho_n}{E_n} > B'_n$ choose to be *converted contributors* who perform model training with a positive amount of local data x^{coop} , and (iii) the contributors at the NE of Game 1 still choose to be *contributors*.

Similar as Theorem 3, Theorem 4 shows that compared with the NE of Game 1 in Theorem 1, we have the following improved results under the cooperative participation strategy. First, the number of free riders is reduced from m to $l - 1$, where $l - 1 \leq m$. Second, the amount of local data chosen by the converted contributors increases from 0 to $x^{coop} > 0$. The following corollary shows that under certain conditions, the cooperative strategy can reduce the number of free riders to 0.

Corollary 3. Under the cooperative strategy in Theorem 4, no client chooses to be a free rider if

$$\frac{\rho_1}{E_1} > B'_1 \triangleq \frac{2\sqrt{G((N-m)D)^3}}{m}.$$

Furthermore, all clients choose to be contributors, i.e., $x_n^{coop} = D, \forall n \in \mathcal{N}$, if

$$\frac{\rho_1}{E_1} > O'_1 \triangleq \frac{D\sqrt{G}}{\frac{1}{\sqrt{(N-m)D}} - \frac{1}{\sqrt{ND}}}.$$

Similar as Corollary 2, Corollary 3 shows that when client 1's ratio $\frac{\rho_1}{E_1}$ is high, all clients choose to be (converted/partial) contributors under the cooperative participation strategy, and there is no free rider in cross-silo FL.

In summary, Theorems 3 and 4 characterize the cooperative participation strategies that can minimize the number of free riders while increasing the amount of local data for model training in cross-silo FL, for both cases of the NE of Game 1 in Theorem 1. We next discuss how to enforce the cooperative participation strategy as the SPNE of Game 2 by a punishment strategy.

4.3 Subgame Perfect Nash Equilibrium

In this section, we aim to derive the condition under which clients behave according to the cooperative participation strategy \mathbf{x}^{coop} at the SPNE of Game 2.

In the infinitely repeated game, we can enforce the cooperative participation strategy \mathbf{x}^{coop} as the SPNE by a punishment strategy \mathbf{x}^{pun} . In this paper, we adopt Friedman punishment [43], where clients play the NE of the stage game SPFL in Theorem 1 if any client deviates from the cooperative participation strategy (i.e., $\mathbf{x}^{pun} = \mathbf{x}^*$). Note that although the cooperative participation strategy can reduce the cost of each client in each time slot, i.e., $F_n(\mathbf{x}^{coop,t}) < F_n(\mathbf{x}^{*,t}), \forall n \in \mathcal{N}$, it is not an equilibrium strategy of the selfish participation game SPFL. In other words, given other clients' strategies under \mathbf{x}^{coop} , a client who is a (converted/partial) contributor has the incentive to decrease its chosen amount of local data for model training to reduce its cost. To punish the client who deviates from the cooperative participation strategy, other clients resort to the NE of the stage game SPFL.

We use the one-stage deviation principle [44] to characterize clients' behaviors at the SPNE. Specifically, each client $n \in \mathcal{N}$ plays the cooperative participation strategy at the SPNE, if its long-term discounted total cost under cooperation is no larger than that under deviation, i.e.,

$$\sum_{t=0}^{\infty} \delta_n^t F_n(\mathbf{x}^{coop}) \leq F_n(\mathbf{x}_n^{least}) + \sum_{t=1}^{\infty} \delta_n^t F_n(\mathbf{x}^{pun}). \quad (18)$$

Here \mathbf{x}_n^{least} is the participation strategy profile under which other clients keep cooperation while client n deviates to minimize its own cost in one time slot. Hence we have

$$\delta_n \geq \delta_n^{th}(\mathbf{x}^{coop}) \triangleq \frac{F_n(\mathbf{x}^{coop}) - F_n(\mathbf{x}_n^{least})}{F_n(\mathbf{x}^{pun}) - F_n(\mathbf{x}_n^{least})}, \quad (19)$$

where $\delta_n^{th}(\mathbf{x}^{coop})$ is the threshold discount factor of client n to play the cooperative strategy profile \mathbf{x}^{coop} . If each client $n \in \mathcal{N}$ is patient enough, i.e., the discount factor satisfies $\delta_n \geq \delta_n^{th}(\mathbf{x}^{coop})$, all clients will play the cooperative participation strategy at the SPNE. For simplicity, we assume that all clients have the same discount factor, i.e., $\delta_n = \delta, \forall n \in \mathcal{N}$, as in [45].

Next we show how to enforce the cooperative strategies derived in Theorems 3 and 4 as the SPNE in the repeated game.

Theorem 5. Consider the following strategy profile: all clients choose the cooperative participation strategy x^{coop} (calculated in Theorems 3 or 4) until a client deviates, and then all clients play the NE x^* of the stage game SPFL (calculated in Theorem 1) in all future time slots. Such a strategy profile is the SPNE of Game 2 if

$$\delta \geq \delta^{th}(x^{coop}) \triangleq \max\{\delta_n^{th}(x^{coop}) : \forall n \in \mathcal{N}\}. \quad (20)$$

Proof. See Appendix G in the supplementary material, available online. \square

Theorem 5 shows that when clients are patient enough, i.e., $\delta \geq \delta^{th}(x^{coop})$, clients play the cooperative strategy x^{coop} at the SPNE, and no client has the incentive to decrease its chosen amount of local data.

Note that the cooperative strategy x^{coop} characterized in Theorems 3 and 4 is not unique, since converted contributors can choose any cooperative participation level $x^{coop} \leq x_l^{th}$. Therefore, the SPNE in Theorem 5 is not unique. Next we design an algorithm to calculate the optimal SPNE that can minimize the number of free riders while maximizing the amount of local data for model training.

4.4 Optimal SPNE

In this section, we aim to derive the optimal SPNE that can minimize the number of free riders while maximizing the amount of local data for model training. In the following, we first define the optimal SPNE of Game 2, and then present the procedures to compute the optimal SPNE.

We first analyze the optimal SPNE $x^{os} = \{x_n^{os} : \forall n \in \mathcal{N}\}$ for the case where the critical client k exists at the NE of Game 1 (as shown in Case I in Theorem 1). Since the optimal SPNE minimizes the number of free riders, it follows the solution structure x^{coop} shown in (11) in Theorem 3, and can be written as

$$x_n^{os} = \begin{cases} 0, & \text{if } n < l; \\ x_{cc}^{os}, & \text{if } l \leq n < k; \\ x_k^*, & \text{if } n = k; \\ D, & \text{if } n > k. \end{cases} \quad (21)$$

Here x_{cc}^{os} is converted contributors' strategy under the optimal SPNE which satisfies $0 \leq x_{cc}^{os} \leq x_l^{th}$, and x_l^{th} is defined in (12), (13). We can see from (21) that only converted contributors' strategy x_{cc}^{os} in the optimal SPNE x^{os} is unknown.

Now we derive x_{cc}^{os} . Since x^{os} is the SPNE of Game 2, it satisfies the one-stage deviation principle described in Theorem 5, i.e., $\delta \geq \delta^{th}(x^{os})$. Furthermore, the optimal SPNE maximizes the amount of local data for model training, i.e., $(k-l)x_{cc}^{os} + x_k^* + (N-k)D$. In summary, x_{cc}^{os} is the optimal solution to the following optimization problem with respect to x_{cc}

$$\begin{aligned} \max \quad & (k-l)x_{cc} + x_k^* + (N-k)D \\ \text{s.t.} \quad & 0 \leq x_{cc} \leq x_l^{th}, \\ & \delta \geq \delta^{th}(x^{os}). \end{aligned} \quad (22)$$

Although the above optimization problem has a simple linear objective with respect to x_{cc} , we cannot derive the closed-form expression for the optimal solution x_{cc}^{os} since the constraint $\delta \geq \delta^{th}(x^{os})$ is complicated due to the following two reasons. First, similar to (20), $\delta^{th}(x^{os})$ takes the maximum value among $\delta_n^{th}(x^{os})$, $\forall n \in \mathcal{N}$, and the max operator is

a non-differentiable function. Second, similar to (19), $\delta_n^{th}(x^{os})$ involves not only x^{os} but also x_n^{least} , and hence the function $\delta_n^{th}(x^{os})$ changes with x_{cc} in a highly non-linear manner (as shown in Appendix G in the supplementary material, available online). In order to find the optimal solution x_{cc}^{os} , we can use greedy algorithms to solve the univariate optimization problem (22).

Algorithm 2. An Algorithm to Calculate the Optimal SPNE of Game 2

Input: $\rho_n, E_n, C_n, \forall n \in \mathcal{N}, G, N, D, p$

Output: The optimal SPNE $x^{os} = \{x_n^{os} : \forall n \in \mathcal{N}\}$

- 1: **if** $\exists k \in \mathcal{N}$ such that $(N-k)D \leq h_k \leq (N+1-k)D$ **then**
- 2: Find the client l where
 $l = \min\{n \in \mathcal{N} : 1 \leq n \leq k-1, \frac{\rho_n}{E_n} > B_n\};$
- 3: Compute the maximum amount of local data x_l^{th} that converted contributors can choose by solving (13);
- 4: Solve the optimization problem (22);
- 5: Announce to each client $n \in \mathcal{N}$ the optimal SPNE:

$$x_n^{os} = \begin{cases} 0, & \text{if } n < l; \\ x_{cc}^{os}, & \text{if } l \leq n < k; \\ x_k^*, & \text{if } n = k; \\ D, & \text{if } n > k. \end{cases}$$

- 6: **else if** $\exists m \in \mathcal{N}$ such that $h_m < (N-m)D < h_{m+1}$
- 7: Find the client l where
 $l = \min\{n \in \mathcal{N} : 1 \leq n \leq m, \frac{\rho_n}{E_n} > B'_n\};$
- 8: Compute the maximum amount of local data x_l^{th} that converted contributors can choose by solving (17);
- 9: Solve the optimization problem (24);
- 10: Announce to each client $n \in \mathcal{N}$ the optimal SPNE:

$$x_n^{os} = \begin{cases} 0, & \text{if } n < l; \\ x_{cc}^{os}, & \text{if } l \leq n \leq m; \\ D, & \text{if } n > m. \end{cases}$$

11: **end**

Next we analyze the optimal SPNE $x^{os} = \{x_n^{os} : \forall n \in \mathcal{N}\}$ for the case where the critical client k does not exist at the NE of Game 1 (as shown in Case II in Theorem 1). Similarly, the optimal SPNE follows the solution structure x^{coop} shown in (15) in Theorem 4, and can be written as

$$x_n^{os} = \begin{cases} 0, & \text{if } n < l; \\ x_{cc}^{os}, & \text{if } l \leq n \leq m; \\ D, & \text{if } n > m. \end{cases} \quad (23)$$

Here x_{cc}^{os} is converted contributors' strategy under the optimal SPNE which satisfies $0 \leq x_{cc}^{os} \leq x_l^{th}$, and x_l^{th} is defined in (16), (17). We can see from (23) that only converted contributors' strategy x_{cc}^{os} in the optimal SPNE x^{os} is unknown. We can derive x_{cc}^{os} by solving the following optimization problem with respect to x_{cc}

$$\begin{aligned} \max \quad & (m-l+1)x_{cc} + (N-m)D \\ \text{s.t.} \quad & 0 \leq x_{cc} \leq x_l^{th}, \\ & \delta \geq \delta^{th}(x^{os}). \end{aligned} \quad (24)$$

Similarly, we can use greedy algorithms to solve the above univariate optimization problem.

Algorithm 2 shows the procedures for the central server to calculate and announce the optimal SPNE of Game 2.¹⁰ To calculate the optimal SPNE of Game 2, the central server first needs to check whether the critical client k exists (Line 1 and Line 6 in Algorithm 2). Then the central server calculates the minimum number of free riders in cross-silo FL under the optimal SPNE, which depends on each client's valuation-computation ratio $\frac{\rho_n}{E_n}$ and the value of B_n or B'_n (Line 2 and Line 7 in Algorithm 2). The central server then calculates the maximum amount of local data x_l^{th} that the converted contributors can choose under the optimal SPNE, by solving (13) or (17) (Line 3 and Line 8 in Algorithm 2). Finally the central server solves the optimization problem (22) or (24) and announces the optimal SPNE strategy x_n^{os} to each client $n \in \mathcal{N}$ (Lines 4-5 and Lines 9-10 in Algorithm 2).

Algorithm 2 has a linear complexity of $\mathcal{O}(D)$. More specifically, checking whether the critical client k exists (Line 1 and Line 6) has a complexity of $\mathcal{O}(N)$. Calculating the number of free riders (Line 2 and Line 7) has a complexity of $\mathcal{O}(N)$. Calculating the maximum amount of local data x_l^{th} (Line 3 and Line 8) has a complexity of $\mathcal{O}(\log D)$. Solving the optimization problem (Line 4 and Line 9) has a complexity of $\mathcal{O}(D)$. Since the number of clients in cross-silo FL is always small, we usually have $N < D$ [4]. Therefore, the overall complexity of Algorithm 2 is $\mathcal{O}(D)$.

5 SIMULATION RESULTS

In this section, we first evaluate the performance of our proposed optimal SPNE and show its scalability. We then compare the optimal SPNE with two related methods.

We conduct simulations on the dataset MNIST using FedAvg [11]. The dataset MNIST contains 60000 figures of handwritten digits, which has been widely used in many existing works on FL, e.g., [46]. We assume that there are $N = 100$ clients since the number of clients in cross-silo FL is always small [4]. Because clients in cross-silo FL are usually companies or organizations which have plenty of data, we assume that each client has a local data set with $D = 10000$ local data samples randomly chosen from MNIST [47]. We assume that the number of iterations in one cross-silo FL process is $G = 50$ under which the FedAvg algorithm can converge [48]. For simplicity, we assume that the computation cost coefficient of each client is $E = 0.985 \times 10^{-8}$ CNY.¹¹ We assume that the valuation for global model accuracy of client n is $\rho_n = n \times \frac{100}{N}$, and the discount factor of each client is $\delta = 0.8$.

5.1 Performance and Scalability

In this subsection, we evaluate the performance of our proposed optimal SPNE in terms of two important metrics: the number of reduced free riders N_f and the total data contribution ratio R_d . The number of reduced free riders is the difference between the number of free riders at the NE of the stage game and the number of free riders at the proposed

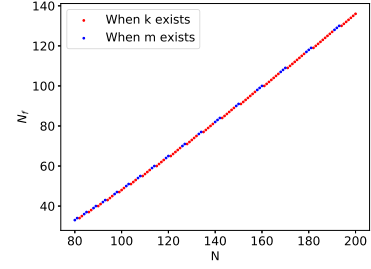


Fig. 3. N_f under different N .

optimal SPNE. The total data contribution ratio is the ratio between the total amount of local data for model training at the optimal SPNE and that at the NE of the stage game, i.e., $R_d = \sum_{n \in \mathcal{N}} x_n^{os} / \sum_{n \in \mathcal{N}} x_n^*$. To show the scalability of our proposed SPNE, we will show how the above two metrics change with the number of clients N and the size of each client's local data set D .

The Impact of the Number of Clients N . We show how the number of clients N affects the number of reduced free riders N_f and the data contribution ratio R_d in Figs. 3 and 4 respectively.

Fig. 3 shows that the number of reduced free riders increases with N . In Fig. 3, the red dots represent the case where the critical client k exists, and the blue dots represent the case where the critical client k does not exist. As N increases, at the NE of the stage game SPFL, the global model trained by the contributors with high valuation-computation ratios is good enough, and hence the contributors with low valuation-computation ratios may become free riders to reduce their total costs. Thus more clients will choose to be free riders with the increase of N . Our proposed optimal SPNE can convert most free riders into *converted contributors* and thus solves the free-rider problem effectively even when N is large.

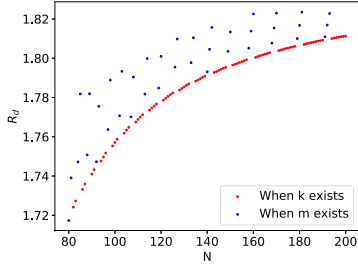
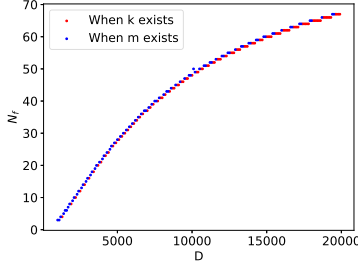
Fig. 4 shows how the data contribution ratio R_d changes with N . Specifically, when the critical client k exists, the data contribution ratio R_d generally increases with N . As N increases, the optimal SPNE can make the majority of free riders choose to be converted contributors and thus increase the data contribution ratio. When k does not exist, under the same number of free riders, the data contribution ratio decreases with N (the blue dots). The reason is that under the same number of free riders, a larger N indicates more contributors, and hence a larger amount of local data for model training, which leads to a global model with a higher model accuracy at the NE of stage game. Therefore, the converted contributors under the optimal SPNE choose a smaller amount of local data for model training to reduce the computation cost. Our proposed optimal SPNE can increase the amount of total data for model training compared with the NE of the stage game, even when N is large.

In summary, when more clients participate in cross-silo FL, the selfish participation behavior leads to more free riders. Our proposed optimal SPNE can effectively reduce the number of free riders by up to 99.3%,¹² and increase the amount of local data

¹⁰. Note that the change in the number of clients participating in the federated learning process due to battery power and occasional disconnections in a mobile computing environment will not affect Algorithm 2.

¹¹. According to Amazon Lambda Pricing [49], each request is charged 1.36×10^{-6} CNY, and the price per millisecond is 1.42×10^{-8} CNY. Our assumption is roughly consistent with [49].

¹². In Fig. 3, when $N = 200$, we can calculate that $k = 138$ and $l = 2$. In other words, there are 137 free riders at the NE of the stage game while there is only one free rider at the optimal SPNE. We reduce the number of free riders by 99.3%, i.e., $\frac{137-1}{137} = 99.3\%$.

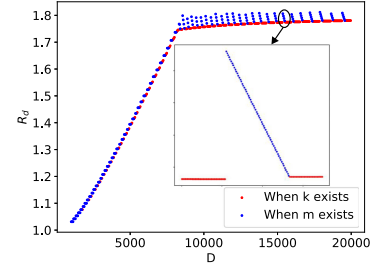

 Fig. 4. R_d under different N .

 Fig. 5. N_f under different D .

chosen by clients to perform model training by up to 82.3%.¹³ Our proposed optimal SPNE performs well even when N is large, which shows the scalability of our work.

The impact of the size of each client's local data set D . We show how the parameter D affects the number of reduced free riders N_f and the data contribution ratio R_d in Figs. 5 and 6 respectively.

Fig. 5 shows that the number of reduced free riders increases with D . When D is small, to achieve a good global model with a high model accuracy requires many clients to choose to be contributors to perform model training, and hence there are few free riders at the NE of the stage game. When D is large, a small number of contributors can train a good global model since each client has a large amount of local data to perform model training, and hence there are many free riders at the NE of the stage game. Our proposed optimal SPNE can convert most free riders into *converted contributors* and thus solves the free-rider problem effectively, even when D is large.

Fig. 6 shows how the data contribution ratio R_d changes with D . Specifically, the data contribution ratio R_d generally increases with D . When D is small, the curve increases quickly with D , and when D is large, R_d increases slowly in a zigzag pattern. When D is small, there are few free riders at the NE of the stage game, and hence there is no much room to increase the amount of local data for model training. As D increases, under the optimal SPNE, converted contributors choose a positive amount of local data to perform model training, which increases the amount of chosen data by up to 81.2%. The subfigure in Fig. 6 shows the zigzag shape of the curve under the same number of free riders. Note that the cooperative strategy of the critical


 Fig. 6. R_d under different D .

client k is x_k^* which decreases with D as calculated in Theorem 1, while the cooperative strategy x_k^{coop} of the converted contributors increases with D . Hence the total amount of local data remains unchanged at the beginning of the curve. When k does not exist, under the same number of free riders, the data contribution ratio decreases with D . The reason is that a larger D leads to less accuracy loss, so the converted contributors will use less data for training to reduce the computation cost. Our proposed optimal SPNE can increase the amount of total data for model training compared with the NE of the stage game, even when D is large.

In summary, when each client in cross-silo FL has more local data samples, the selfish participation behavior leads to more free riders. Our proposed optimal SPNE can effectively reduce the number of free riders by up to 98.5%, and increase the data contribution by up to 81.2%. Our proposed optimal SPNE performs well even when D is large, which shows its scalability.

We also evaluate the performances of our proposed optimal SPNE under other system parameters such as the number of iterations G , the computation cost coefficient E , and the discount factor δ . Due to space limit, we put detailed discussions in Appendix H, available in the online supplemental material.

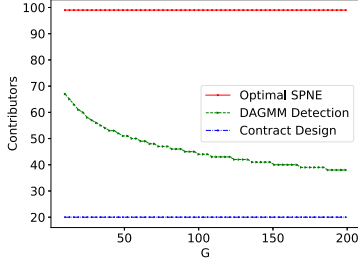
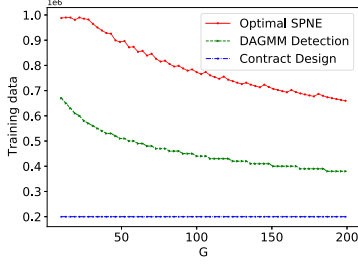
5.2 Comparisons With Related Methods

In this subsection, we compare our proposed solution with two related methods. One is a free-rider detection method called DAGMM proposed in [50], which can effectively detect free riders and only aggregate the updates of (partial) contributors at the server. The other is a contract-based incentive mechanism for FL proposed in [16]. In the following, we show the comparisons between our method and the above two methods under different system parameters (i.e., the number of iterations G and the discount factor δ).¹⁴

Figs. 7 and 8 show the comparisons of the three methods under different values of G . A larger G means more iterations in one cross-silo FL process, in which case clients can achieve a global model with a higher model accuracy. Fig. 7 shows that among the three methods, our method achieves the largest number of contributors under different values of G . Specifically, under our method, the number of contributors remains unchanged and almost all clients participate in the training. Under the DAGMM detection method, the number of contributors decreases with G . Under the contract design method, only 20% clients participate in the training. The reason is that under a larger G , the global

13. In Fig. 4, when $N = 181$, we can calculate that $m = 120$. Furthermore, the total amount of local data for model training at the optimal SPNE is 1112299, and the total amount of local data for model training at the NE of the stage game is 610000. In this case, the total data contribution ratio is $\frac{1112299}{610000} = 1.823$. Equivalently, the amount of local data chosen by clients to perform model training is increased by 82.3%.

14. We also show the comparisons under different values of N , D and E in Appendix H, available in the online supplemental material.

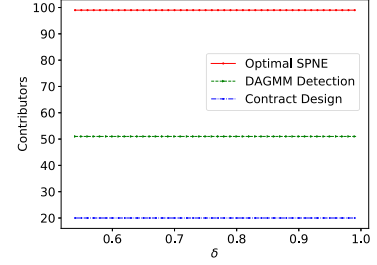
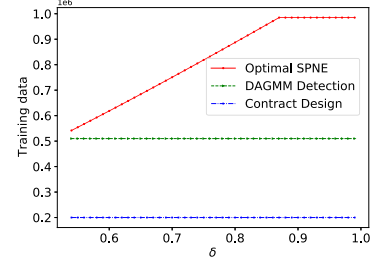
Fig. 7. The number of contributors under different G .Fig. 8. The total amount of training data under different G .

model trained by contributors with high valuation-computation ratios is good enough, and hence contributors with low valuation-computation ratios for the global model accuracy may change to be free riders. Therefore, the number of contributors under the DAGMM detection method decreases with G . Under the contract design method, since the change in G does not affect the number of clients that are provided with a positive contract item, the number of contributors remains unchanged. Our proposed method can effectively motivate almost all free riders to become converted contributors.

Fig. 8 shows that our method has a larger amount of training data compared with the other two methods. Specifically, under our method, the total amount of training data first remains unchanged, and then gradually decreases. The reason is that a smaller G leads to a smaller threshold discount factor of clients. So when G is small, converted contributors will participate in training with all local data. As G increases, converted contributors can use only part of local data in training. Under the DAGMM detection method, since the number of contributors decreases with G , the total amount of training data also decreases with G . Under the contract design method, since the change in G does not affect the number of contributors, the total amount of training data remains unchanged.

Figs. 9 and 10 show the comparisons of the three methods under different values of δ . Fig. 9 shows that the number of contributors under the three methods does not change with the discount factor δ . The reason is that the number of free riders depends on the valuation-computation ratios $\{\frac{p_n}{E_n} : \forall n \in \mathcal{N}\}$, the bounds $\{B_n : \forall n \in \mathcal{N}\}$, and the local data set sizes $\{D_n : \forall n \in \mathcal{N}\}$, and is independent of δ . Our method can motivate almost all free riders to become converted contributors, and hence our method has distinct advantage over the other two methods.

Fig. 10 shows that our method has a larger amount of training data compared with the other two methods.

Fig. 9. The number of contributors under different δ .Fig. 10. The total amount of training data under different δ .

Specifically, under our method, the total amount of training data first increases with δ , and then remains to be a constant when δ is larger than 0.87. A larger discount factor δ indicates that converted contributors are more patient, and hence they are more willing to choose a larger amount of local data to reduce the long-term discounted total cost. When converted contributors are patient enough, i.e., $\delta \geq 0.87$, they will perform model training with all their local data. Under the DAGMM detection method and the contract design method, the numbers of contributors remain unchanged, and the contributors under these two methods participate in training with all their local data. Thus the amounts of training data of these two methods also remain unchanged.

6 CONCLUSION

In this work, we analyze clients' selfish participation behaviors in cross-silo FL. We model the interactions among clients in a single cross-silo FL process as a selfish participation game and derive its unique Nash equilibrium. We show that clients' selfish participation behaviors lead to free riders at equilibrium. The existence of free riders hampers clients' long-term participation in cross-silo FL. We model clients' interactions in the long-term cross-silo FL processes as a repeated game. We derive the optimal SPNE that can minimum the number of free riders while maximizing the amount of local data for model training. Simulation results show that our derived optimal SPNE can effectively reduce the number of free riders by up to 99.3% and increase the amount of data for training by up to 82.3%. For future work, it is interesting to analyze the interactions among clients in the incomplete information scenario where clients do not know the participation strategies of other clients or in the asymmetric information scenario where the valuation for global model accuracy and the computation cost coefficient are the private information of each client.

REFERENCES

- [1] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan./Feb. 2018.
- [2] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.
- [3] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on Non-IID data," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 7865–7873.
- [4] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, pp. 1–210, 2021.
- [5] L. Insurance, "Swiss Re and Chinese firm WeBank partner to explore use of AI in reinsurance." Accessed: Apr. 11, 2021. [Online]. Available: <https://www.lifeinsuranceinternational.com/news/swiss-re-webank/>
- [6] E. research project, "Mammogram assessment with NVIDIA Clara federated learning." Accessed: Apr. 11, 2021. [Online]. Available: <https://blogs.nvidia.com/blog/2020/04/15/federated-learning-mammogram-assessment/>
- [7] C. Hale, "The MELLODDY project." Accessed: Apr. 11, 2021. [Online]. Available: <https://www.fiercebiotech.com/special-report/melloddy-project>
- [8] M. Tang and V. W. S. Wong, "An incentive mechanism for cross-silo federated learning: A public goods perspective," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
- [9] P. J. Miller and D. M. Chin, "Using monthly data to improve quarterly model forecasts," *Federal Reserve Bank Minneapolis Quart. Rev.*, vol. 20, pp. 16–28, 1996.
- [10] N. Van Leersum et al., "The dutch surgical colorectal audit," *Eur. J. Surg. Oncol.*, vol. 39, no. 10, pp. 1063–1070, 2013.
- [11] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
- [12] Z. Zhong et al., "P-FedAvg: Parallelizing federated learning with theoretical guarantees," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
- [13] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
- [14] X. Mo and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," 2020, *arXiv:2003.00199*.
- [15] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
- [16] 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.
- [17] P. Sun et al., "Pain-FL: Personalized privacy-preserving incentive for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3805–3820, Dec. 2021.
- [18] Y. Zhan and J. Zhang, "An incentive mechanism design for efficient edge learning by deep reinforcement learning approach," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 2489–2498.
- [19] R. Zeng, S. Zhang, J. Wang, and X. Chu, "FMore: An incentive scheme of multi-dimensional auction for federated learning in MEC," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 278–288.
- [20] Y. Deng et al., "FAIR: Quality-aware federated learning with precise user incentive and model aggregation," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
- [21] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, "A survey of incentive mechanism design for federated learning," *IEEE Trans. Emerg. Topics Comput.*, early access, Mar. 03, 2021, doi: [10.1109/TETC.2021.3063517](https://doi.org/10.1109/TETC.2021.3063517).
- [22] W. J. Baumol, "Welfare economics and the theory of the state," in *The Encyclopedia of Public Choice*. Berlin, Germany: Springer, 2004, pp. 937–940.
- [23] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 1010–1019, May 2006.
- [24] J. Lin, M. Du, and J. Liu, "Free-riders in federated learning: Attacks and defenses," 2019, *arXiv:1911.12560*.
- [25] J. Huang, R. Talbi, Z. Zhao, S. Boucchenak, L. Y. Chen, and S. Roos, "An exploratory analysis on users' contributions in federated learning," 2020, *arXiv:2011.06830*.
- [26] Y. Fraboni, R. Vidal, and M. Lorenzi, "Free-rider attacks on model aggregation in federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 1846–1854.
- [27] Y. Chen, X. Yang, X. Qin, H. Yu, B. Chen, and Z. Shen, "FOCUS: Dealing with label quality disparity in federated learning," 2020, *arXiv:2001.11359*.
- [28] M. A. Heikkilä, A. Koskela, K. Shimizu, S. Kaski, and A. Honkela, "Differentially private cross-silo federated learning," 2020, *arXiv:2007.05553*.
- [29] U. Majeed, L. U. Khan, and C. S. Hong, "Cross-silo horizontal federated learning for flow-based time-related-features oriented traffic classification," in *Proc. 21st Asia-Pacific Netw. Operations Manage. Symp.*, 2020, pp. 389–392.
- [30] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proc. USENIX Annu. Tech. Conf.*, 2020, pp. 493–506.
- [31] N. Ding, Z. Fang, and J. Huang, "Incentive mechanism design for federated learning with multi-dimensional private information," in *Proc. 18th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw.*, 2020, pp. 1–8.
- [32] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 165–202, 2012.
- [33] X. Zhang, F. Li, Z. Zhang, Q. Li, C. Wang, and J. Wu, "Enabling execution assurance of federated learning at untrusted participants," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1877–1886.
- [34] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 661–670.
- [35] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1387–1395.
- [36] G. H. Lee and S.-Y. Shin, "Federated learning on clinical benchmark data: Performance assessment," *J. Med. Internet Res.*, vol. 22, no. 10, 2020, Art. no. e20891.
- [37] M. Diamanti, G. Fragkos, E. E. Tsiropoulou, and S. Papavassiliou, "Unified user association and contract-theoretic resource orchestration in NOMA heterogeneous wireless networks," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1485–1502, Sep., 2020, doi: [10.1109/OJCOMS.2020.3024778](https://doi.org/10.1109/OJCOMS.2020.3024778).
- [38] H. Sun, X. Ma, and R. Q. Hu, "Adaptive federated learning with gradient compression in uplink NOMA," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16 325–16 329, Dec. 2020.
- [39] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2982–2997, Dec. 2018.
- [40] S. Shakkottai and R. Srikant, "Economics of network pricing with multiple ISPs," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1233–1245, Dec. 2006.
- [41] Q. Ma, J. Huang, T. Başar, J. Liu, and X. Chen, "Reputation and pricing dynamics in online markets," *IEEE/ACM Trans. Netw.*, vol. 29, no. 4, pp. 1745–1759, Aug. 2021.
- [42] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4615–4625.
- [43] J. W. Friedman, "A non-cooperative equilibrium for supergames," *Rev. Econ. Studies*, vol. 38, no. 1, pp. 1–12, 1971.
- [44] E. Hendon, H. J. Jacobsen, and B. Sloth, "The one-shot-deviation principle for sequential rationality," *Games Econ. Behav.*, vol. 12, no. 2, pp. 274–282, 1996.
- [45] G. J. Mailath and L. Samuelson, *Repeated Games and Reputations: Long-Run Relationships*. London, U.K.: Oxford Univ. Press, 2006.
- [46] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on Non-IID data," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–26.
- [47] S. Francis, I. Tenison, and I. Rish, "Towards causal federated learning for enhanced robustness and privacy," 2021, *arXiv:2104.06557*.
- [48] Y. Liu, Y. Kang, X. Zhang, L. Li, Y. Cheng, T. Chen, M. Hong, and Q. Yang, "A communication efficient collaborative learning framework for distributed features," 2019, *arXiv:1912.11187*.
- [49] Amazon, "Amazon lambda pricing." Accessed: Jun. 22, 2021. [Online]. Available: <https://www.amazonaws.cn/en/lambda/pricing/>

- [50] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–19.



Ning Zhang (Student Member, IEEE) received the BE degree from the School of Intelligent Systems Engineering, Sun Yat-sen University (SYSU), Guangzhou, China, in 2020. He is currently working toward the MS degree in the School of Intelligent Systems Engineering, SYSU, Guangzhou, China. His primary research interests include game theory, incentive mechanism, and federated learning.



Qian Ma (Member, IEEE) received the BS degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2012, and the PhD degree from the Department of Information Engineering, Chinese University of Hong Kong, Hong Kong, in 2017. She is an associate professor with the School of Intelligent Systems Engineering, Sun Yat-sen University. She worked as a postdoc research associate with Northeastern University during 2018–2019. Her research interests lie in the field of network optimization and

economics. She is the recipient of the Best Paper Award from the IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt) in 2021 and the recipient of the Best Student Paper Award from the IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt) in 2015.



Xu Chen (Senior Member, IEEE) received the PhD degree in information engineering from the Chinese University of Hong Kong, Hong Kong, in 2012. He is a full professor with Sun Yat-sen University, Guangzhou, China, and the vice director of National and Local Joint Engineering Laboratory. He worked as a postdoctoral research associate with Arizona State University, Tempe, USA, from 2012 to 2014, and a Humboldt scholar fellow with the Institute of Computer Science, University of Goettingen, Germany from 2014 to 2016. He

received the prestigious Humboldt research fellowship awarded by the Alexander von Humboldt Foundation of Germany, 2014 Hong Kong Young Scientist Runner-up Award, 2017 IEEE Communication Society Asia-Pacific Outstanding Young Researcher Award, 2017 IEEE ComSoc Young Professional Best Paper Award, Honorable Mention Award of 2010 IEEE international conference on Intelligence and Security Informatics (ISI), Best Paper Runner-up Award of 2014 IEEE International Conference on Computer Communications (INFOCOM), and Best Paper Award of 2017 IEEE International Conference on Communications (ICC). He is currently an area editor of the *IEEE Open Journal of the Communications Society*, an associate editor of the *IEEE Transactions Wireless Communications*, *IEEE Transactions Vehicular Technology*, and *IEEE Internet of Things Journal*.

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