

Collaboration in Participant-Centric Federated Learning: A Game-Theoretical Perspective

Guangjing Huang¹, Xu Chen¹, Tao Ouyang¹, Qian Ma¹, Lin Chen¹, and Junshan Zhang², *Fellow, IEEE*

Abstract—Federated learning (FL) is a promising distributed framework for collaborative artificial intelligence model training while protecting user privacy. A bootstrapping component that has attracted significant research attention is the design of incentive mechanism to stimulate user collaboration in FL. The majority of works adopt a broker-centric approach to help the central operator to attract participants and further obtain a well-trained model. Few works consider forging participant-centric collaboration among participants to pursue an FL model for their common interests, which induces dramatic differences in incentive mechanism design from the broker-centric FL. To coordinate the selfish and heterogeneous participants, we propose a novel analytic framework for incentivizing effective and efficient collaborations for participant-centric FL. Specifically, we respectively propose two novel game models for contribution-oblivious FL (COFL) and contribution-aware FL (CAFL), where the latter one implements a minimum contribution threshold mechanism. We further analyze the uniqueness and existence for Nash equilibrium of both COFL and CAFL games and design efficient algorithms to achieve equilibrium solutions. Extensive performance evaluations show that there exists free-riding phenomenon in COFL, which can be greatly alleviated through the adoption of CAFL model with the optimized minimum threshold.

Index Terms—Federated learning, game theory, nash equilibrium, collaboration strategy

1 INTRODUCTION

WITH the rapid development of Internet of Things (IoT) and mobile networks, massive volumes of user data are being generated and geographically scattered over the edges of networks [1], [2], [3]. To learn valuable knowledge and information from these massive data, artificial intelligence technology (AI) has been widely used to support machine learning, which dramatically expedites the emergence of many intelligent IoT mobile applications [4]. Traditional machine learning frameworks typically require the uploading local training data to a central server for centralized learning, which suffers from the risk of privacy leaks [5]. To address such critical issue, federated learning (FL) has been proposed by Google as a novel paradigm to train AI model in a privacy-

preserving manner [6], in which the device users learn from their local data and then upload their local model updates to the central server. Since only the local update (e.g., parameter gradients), rather than the local data, is sent to the central server by encrypted communication, the FL enables the device users to preserve data privacy efficiently.

Nevertheless, local model training incurs significant costs (e.g., computation cost or energy consumption) for the involved users in FL. Without a proper incentive mechanism, selfish participants would be reluctant to participate in FL. In terms of incentive mechanism design, the majority of existing studies focus on a broker-centric paradigm for incentivizing participation in FL (Fig. 1a). That is, a central broker publishes a model training task and becomes a model stakeholder (model owner), who attracts participants (workers) to help complete the model training through monetary rewards via economic mechanisms such as auction and contract design [7], [8], [9], [10], [11], [12], [13]. In these settings, the participants aim to get more payments from the broker through local model training and uploading.

Along a different line, in many application scenarios, FL training tasks are not generated by a central broker, but initiated by a group of users who themselves are also the participants in the FL [14], [15], in order to collectively train a common FL model. In such participant-centric FL (Fig. 1b), the participants are the model stakeholders and they have the common interest of obtaining a high-quality FL model via mutually-beneficial knowledge sharing. In this case, the FL server works as a coordinator to assist the FL procedure. For example, for smart home applications, users of different families can collectively boost the performance of in-home healthcare AI service via FL [16].

However, how to design a proper incentive mechanism for participant-centric FL is much less understood in the

- Guangjing Huang, Xu Chen, Tao Ouyang, and Lin Chen are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China. E-mail: {huanggj27, ouyt9}@mail2.sysu.edu.cn, {chenxu35, chenlin69}@mail.sysu.edu.cn.
- Qian Ma is with the School of Intelligent System Engineering, Sun Yat-sen University, Guangzhou 510006, China. E-mail: maqian25@mail.sysu.edu.cn.
- Junshan Zhang is with the ECE Department, University of California, Davis, Davis, CA 95616 USA. E-mail: jzhang@ucdavis.edu.

Manuscript received 21 February 2022; revised 7 July 2022; accepted 20 July 2022. Date of publication 27 July 2022; date of current version 3 October 2023.

This work was supported in part by the National Science Foundation of China under Grants 61972432, U20A20159, 62172455, and 62002399, in part by Guangdong Basic and Applied Basic Research Foundation under Grants 2021B151520008 and 2022A1515012309, in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X355, and in part by Pearl River Talent Program under Grants 2017GC010465 and 2019QN01X140. (Corresponding author: Xu Chen.)

This article has supplementary downloadable material available at <https://doi.org/10.1109/TMC.2022.3194198>, provided by the authors. Digital Object Identifier no. 10.1109/TMC.2022.3194198

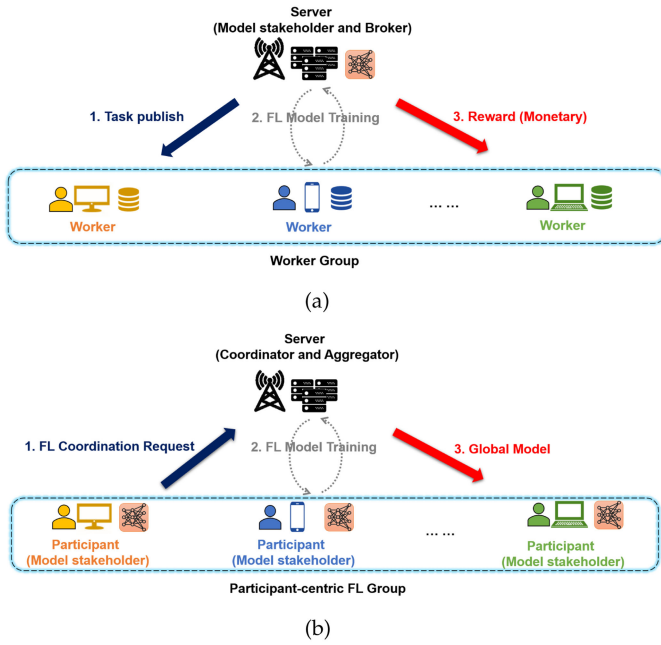


Fig. 1. An illustration of (a) broker-centric Federated Learning (b) participant-centric Federated Learning.

literature [17]. Given that all participants are selfish to act for their own benefits and are also heterogeneous in various perspectives (e.g., valuation on the FL model, data sizes and computing capability), how to obtain an effective strategy for stimulating mutually-beneficial collaboration among the participants is challenging. We should emphasize that, for broker-centric FL, users' heterogeneous cost can be well characterized and compensated with the monetary incentive by the centralized broker via classic two-sided economic mechanisms such as auctions and contracts. However, for participant-centric FL, obtaining a common FL model is the major incentive for the users, but they have usually heterogeneous cost and valuations with complex interaction structures, and their marginal benefits/contributions for each other in the FL model training are very hard to quantify. Hence designing monetary incentive mechanisms that can ensure mutually-satisfactory and trustworthy payment transferring among the users as rewards are generally prohibitive in practice, and thus monetary-free incentive mechanisms with lightweight implementation complexity are much more desired for participant-centric FL. Specifically, we need to address the following key issues for incentivizing collaboration in participant-centric FL.

First of all, a monetary-free incentive-driven collaboration strategy for participant-centric FL is essential, because otherwise the participants acting for their own benefits may deviate from the collaboration strategy. This requirement motivates a key issue: *How can we model the heterogeneous participants' selfish behaviors and derive an effective collaboration strategy such that each participant has the incentive to follow without external monetary rewards?*

Second, since the trained FL model can be shared among the participants, some participants may benefit from FL but contribute nothing, namely "free-riding". The free-riding phenomenon leads to great unfairness in participant-centric

FL. This motivates the second key issue: *How can we design an efficient mechanism to thwart the free-riding behaviors in participant-centric FL?*

Last but not least, given that participant-centric nature of FL in our setting, how to attain a high global efficiency with a low implementation complexity would be a key challenge. This motivates the third key issue: *How can we maximize the total utility of the participants while maintaining the self-stability of FL in a lightweight manner?*

To address the above two key issues, we will leverage the game theoretic modeling approach for developing a comprehensive analytical framework for participant-centric FL. We take into account users' selfish behaviors and heterogeneous characteristics, in order to derive effective collaboration strategies with the desired properties of incentive-driven collaboration, free-riding mitigation and global efficiency boosting for participant-centric FL with a simple minimum contribution threshold mechanism. The contributions of this paper are summarized as follows:

- **Contribution-oblivious FL (COFL) game analysis:** To understand participants' selfish behaviors, we first propose a participant-centric FL game called COFL to enable participants to train a shared federated model collectively without imposing the contributing requirements. We show that the COFL game admits a unique Nash equilibrium under some regularity conditions, and also devise an algorithm to achieve it. However, our findings reveal that the free-riding phenomenon exists in COFL, which would lead to the critical issue of unfairness and greatly harm participants' motivations for collaboration in participant-centric FL.
- **Contribution-aware FL (CAFL) game analysis:** To alleviate the free-riding behaviors in participant-centric FL, we then devise a simple and effective collaboration mechanism and propose a novel enhanced game model of CAFL, where a participant will be excluded from getting the trained FL model if his promised training batchsize does not reach the minimum contribution threshold (i.e., a minimum threshold batchsize mechanism). However, we show that the existence of Nash equilibrium cannot be always guaranteed in CAFL, which may cause instable participating behaviors in FL. Thus, we propose an algorithm to refine the set of participants in CAFL, which can always guarantee to achieve a Nash equilibrium of the refined CAFL game. Furthermore, we boost the global performance by finding the optimal threshold to maximize the total utility of all participants.
- **Extensive performance evaluation:** We finally conduct extensive numerical evaluations with the realistic MNIST and CIFAR10 datasets to verify our theoretical analysis and results. We find that, compared with COFL, CAFL not only effectively alleviates the free-riding phenomenon, but also significantly boost the amount of participation and total utility to a great extent, stimulating more than 90% participants to contribute with a superior performance very closing to the solution of optimal total utilities in most cases.

The rest of this paper is organized as follows. Section 2 introduces FL framework and problem formulation respectively. Contribution-oblivious FL (COFL) is presented in Section 3. We discuss Contribution-aware FL system and algorithm design in Sections 4 and 5 respectively. Numerical result is shown in Section 6. We introduce related work in Section 7 followed by conclusion in Section 8.

2 SYSTEM MODEL

We consider the participant-centric FL formed by multiple device users at the network edge. We adopt the standard federated learning framework in [6], [18] and define a global iteration as follows. During the local training phase, all device participants calculate local gradients based on the their data and the received model. Then the edge server (i.e., central server) aggregates all local gradients from participants (i.e., the device users) and broadcasts new gradients to each participant for next local model updates. This complete update process is called a global iteration. All participants repeat the global iteration until the global model converges. Since model training incurs significant time and energy cost, participants should carefully strike a balance between their valuation on the FL model (i.e., getting the model as rewards) and the training costs. In what follows, we first characterize participants' training costs, and then calculate participants' utilities.

2.1 Participant Cost Model

In a global iteration, a participant's model training cost consists of computation and communication cost.

Computation Cost. The set of K participants is denoted as $\mathcal{K} = \{1, 2, \dots, K\}$. Each participant k owns a local dataset $\mathcal{D}_k = \{(\mathbf{x}_k^1, y_k^1), \dots, (\mathbf{x}_k^{N_k}, y_k^{N_k})\}$, and $|\mathcal{D}_k|$ denotes the data size of participant k , and $\mathbf{x}_k^i$ and y_k^i represent the model features and the Ground truth label of participant k 's i th data respectively. Similar to most existing approaches such as [18], we adopt the mini-batch stochastic gradient descent (SGD) algorithm for local training. Each participant selects a subset of the local dataset, called one batch, to calculate local gradients in one global iteration. The number of data samples in one batch selected by participant k is B_k and the global batchsize is $B = \sum_{k \in \mathcal{K}} B_k$. Obviously, $0 \leq B_k \leq B_k^{\max} \triangleq |\mathcal{D}_k|$. To model a participant's energy consumption, we further define C_k as the average number of CPU cycles for participant k performing local gradient calculation with one data sample. Moreover, f_k is the adopted CPU processing speed (i.e., computing resource) of participant k for local model training. In general, we have $f_k^{\min} \leq f_k \leq f_k^{\max}$ where $f_k^{\min}$ and $f_k^{\max}$ represent participant k 's minimum and maximum computing capabilities, respectively. Based on the above definitions, the energy consumption of local gradients calculation of participant k in one global iteration can be calculated as [19]

$$E_k^{\text{com}} = \frac{1}{2} \alpha C_k B_k f_k^2, \quad (1)$$

where α is a coefficient corresponding to the computing chip architecture. Also, the local training process latency can be calculated as

$$T_k^{\text{com}} = \frac{C_k B_k}{f_k}. \quad (2)$$

Communication Cost. At the phase of data communication, the local gradients are sent to the center server through the multiple channel access technology (e.g., orthogonal frequency division medium access (OFDMA)) once the local model training accomplished. The communication resource allocation optimization is usually determined by the central server instead of participants. Since we focus on analyzing the participants' behaviors in participant-centric FL, we assume that the communication resource allocation for the participants' data transmission is fixed. We also assume that the data size of the model parameter gradients is the same for each participant in each FL iteration (which is usually the case since all the participants train the same FL model). Thus, in this study we hence assume the communication cost is fixed for each participant in the following discussions¹.

2.2 Participant Utility Function

We next turn attention to the participants' utilities in participant-centric FL. In general, all participants aim to obtain a well-trained model with a low loss function (or high model accuracy). In one global iteration, higher model accuracy requires a larger global batchsize, which will further induce a higher training cost (i.e., local energy consumption and latency). In this regard, according to [18], we first define the global loss function and the global loss decrease (model accuracy improvement) as follows:

$$L(\mathbf{w}) = \frac{1}{|\bigcup_{k \in \mathcal{K}} \mathcal{D}_k|} \sum_{k \in \mathcal{K}} |\mathcal{D}_k| L_k(\mathbf{w}, \mathcal{D}_k), \quad (3)$$

$$\Delta L[n] = L(\mathbf{w}[n]) - L(\mathbf{w}[n-1]), \quad (4)$$

where $L_k(\mathbf{w}, \mathcal{D}_k)$ is a general convex loss function of participant k , wherein $\mathbf{w}$ is the machine learning model parameter. $L(\mathbf{w}[n])$ is the loss function of n th global iteration. We consider one global iteration and replace $\Delta L[n]$ with ΔL . We can measure the model accuracy improvement in one global iteration by ΔL which depends on global batchsize. According to [23], the expected difference $f(\mathbf{w}[n]) - f(\mathbf{w}^*)$ is bounded by $O(1/\sqrt{Bn} + 1/n)$, when participants use mini-batch SGD in the IID case, where B is global batchsize and n is the number of global iteration. Given fixed n , the upper bound of expected difference is decreasing and convex function respect to global batchsize B and satisfies diminishing marginal effect. In one global iteration, the model improvement ΔL is dominated by term $\sqrt{B}$ and can be approximately expressed as [18]

$$\Delta L = \xi \sqrt{B} = \xi \sqrt{\sum_{k \in \mathcal{K}} B_k}, \quad (5)$$

1. The communication cost for each participant can be different and this case can be captured by adding a participant-specific constant D_k into the participant's utility function in (6). Nevertheless, since the communication cost D_k is fixed and does not impact participant's decision on the computation resource allocation and batch size selection, we will neglect the fixed communication cost in the following analysis for simplicity. This simplicity is common in many exiting studies related to incentive mechanism design [20], [21], [22] and conducive to directly revealing economic characteristic for mechanism design, since the communication part is independent of FL model performance.

where ξ depends on the model structure. For simplicity, we use $\xi = 1$ in this paper because it is a scaling constant for a given training model. Intuitively, a large amount of training data leads to a good model performance in one global iteration. Based on the above definitions, the utility function for participant k is

$$U_k(B_k, f_k) = \theta_k \ln(1 + \Delta L) - \varphi_k E_k^{com} - \gamma_k T_k^{com}. \quad (6)$$

Here, θ_k describes participant's preference of the FL model (i.e., a larger θ_k implies that participant k has a higher valuation on the model). φ_k and γ_k are weight parameters corresponding to participant k 's energy consumption and latency of local training processing respectively. Note that in (6) we omit the decision variables of other participants for notational simplicity. The utility function (6) captures that a participant should jointly consider the model improvement, energy consumption and local training time. Corresponding to the economic characteristics of the upper bound of expected difference, the term $\theta_k \ln(1 + \Delta L)$ is increasing and concave function respect to global batchsize B and reveals the rule of diminishing marginal returns for model improvement [24]. The physical meaning of the logarithmic function is that the magnitude of FL model's improvement decreases with global batchsize. That is, when a FL model possesses a large global batchsize, participants make the same effort but obtain a low return, which would encourage participants to contribute more when the global batchsize is small. Note that, for ease of exposition, we adopt a specific logarithmic function in this paper. Actually, our theoretical results and analysis still hold for a general function $\theta_k g(B)$ wherein the model accuracy related function $g(B)$ is a second-order differentiable function and satisfies diminishing marginal effect ($g'(B) > 0$, $g''(B) < 0$ and $g'(0) > \frac{A_k}{\theta_k}$). Here, A_k is the unit training cost (which will be discussed later). The discussion on general function is given in Section A in Appendix C in the separate supplementary file, (available online).

Assuming the FL process converges in a finite number of global iterations,² we define the utility function of one global iteration in (6) to estimate a participant's utility throughout the training process. Based on (1), (2), (5), and (6), each participant k aims to maximize his utility function by tuning their local training batchsize and CPU processing speed, which can be calculated as

$$U_k(B_k, f_k) = \theta_k \ln(1 + \sqrt{B_{-k} + B_k}) - \frac{1}{2} \varphi_k \alpha C_k B_k f_k^2 - \gamma_k \frac{C_k B_k}{f_k}, \quad (7)$$

where $B_{-k} = \sum_{i \in \mathcal{K} \setminus \{k\}} B_i$ shows that a participant utility is also influenced by other participants' decisions in terms of the choice of batchsize for local training. For ease of presentation, we predefine $A_k = \frac{1}{2} \varphi_k \alpha C_k f_k^{*2} + \gamma_k \frac{C_k}{f_k^*}$ as the unit

training cost of participant k , where f_k^* is the optimal CPU frequency (which will be discussed later).

Note that in this paper, we assume that all participants are selfish but have no intention of sabotaging the FL model. The security issue of FL model is not the focus of this paper. We also assume that all participants' data are independent and identically distributed (IID). The scenario of non-IID data distributions will be considered in a future work.

In the following, aiming at modeling participants' strategic behaviors in the participant-centric FL, we leverage the game theoretical approach to derive useful insights and devise efficient collaboration strategies accordingly.

3 CONTRIBUTION-OBVIOUS FL GAME

We first consider the contribution-oblivious FL (COFL) game model, in which all participants act on their own benefits to decide the computing resources and data batchsize for local training without imposing any contributing requirements. Due to the heterogeneous nature among participants (e.g., differences in model preference and training cost), forging an incentive-driven collaboration strategy such that all participants are mutually satisfied is non-trivial. We will analyze the Nash equilibrium of COFL game, propose an algorithm to achieve Nash equilibrium in COFL game, and derive some insightful results.

3.1 Game Formulation and Best Response in COFL

We formally define the COFL game as follows:

Game 1 (COFL Game).

- **Players:** The set $\mathcal{K}$ of participants.
- **Strategies:** The chosen batchsize $B_k \in [0, B_k^{max}]$, and computing speed $f_k \in [f_k^{min}, f_k^{max}]$ for each $k \in \mathcal{K}$.
- **Utilities:** The utility $U_k(B_k, f_k)$ for each $k \in \mathcal{K}$.

In the following, we combine the choices of data batchsize B_k and computing speed f_k as a computation strategy $\sigma_k(B_k, f_k) \in \Sigma_k$, where Σ_k is the participant k 's strategy space. Here, we treat B_k as real number, the strategy space is thus a convex hull.

The COFL game reaches Nash equilibrium if and only if none of participants can unilaterally change the strategy to improve his utility. A Nash equilibrium solution is a strategy profile $(\sigma_1^*, \dots, \sigma_K^*)$ such that $\forall k \in \mathcal{K}$, the following inequality holds:

$$U_k(\sigma_1^*, \dots, \sigma_k, \dots, \sigma_K^*) \leq U_k(\sigma_1^*, \dots, \sigma_k^*, \dots, \sigma_K^*), \forall \sigma_k \in \Sigma_k. \quad (8)$$

This implies all participants take the mutually best response strategy simultaneously in the FL model training, i.e.,

$$\hat{\sigma}_k^* = \arg \max_{\sigma_k \in \Sigma_k} U_k(\sigma_1^*, \dots, \sigma_k, \dots, \sigma_K^*). \quad (9)$$

For the ease of practical implementation, we only consider the pure Nash equilibrium in this paper³. To characterize the

2. The number of global iterations for a fixed model accuracy depends on global batchsize. In general, the global batchsize is much larger than the local batchsize of a participant. Thus, we assume that a single participant's decision has little impact on the number of global iterations.

3. We adopt the widely used Nash equilibrium as the result of game analysis. The inefficient cooperation among the participants in Nash equilibrium is the motivation for our subsequent improvements.

existence of Nash equilibrium, we first derive participant's the best response strategy in COFL. Given others' decisions, the best response function of participant k is shown as below.

Proposition 1. The best response strategy σ_k of participant k in COFL is (B_k^*, f_k^*)

$$f_k^* = \left[\sqrt[3]{\frac{\gamma_k}{\varphi_k \alpha}} \right]_{f_k^{\min}}^{f_k^{\max}}, \quad (10)$$

and

$$B_k^* = [\tilde{B}_k]_0^{B_k^{\max}}, \quad (11)$$

where $[X]_Y^Z = \min\{\max\{Y, X\}, Z\}$, $A_k = \frac{1}{2} \varphi_k \alpha C_k f_k^{*2} + \gamma_k \frac{C_k}{f_k}$, and

$$\tilde{B}_k = \left(\sqrt{\frac{1}{4} + \frac{1}{2} \frac{\theta_k}{A_k}} - \frac{1}{2} \right)^2 - B_{-k}. \quad (12)$$

The proof is given in Section A in Appendix A in the separate supplementary file, available online. Since it can be seen from (10) that a participant's optimal CPU frequency is independent of others' strategies, the key issue of the COFL game is how coordinate participants' data batchsizes to achieve a Nash equilibrium.

3.2 Nash Equilibrium in COFL

In this subsection, we focus on finding the Nash equilibrium of COFL game. Specifically, according to best response functions in Proposition 1 above, we first capture the characteristic of the Nash equilibrium of the COFL game (Theorem 1), and based on which, we then obtain equivalent form of Nash equilibrium (Theorem 2). We next show the existence and uniqueness of the Nash equilibrium of the COFL game (Theorem 3). Finally, the equilibrium finding algorithm is given in Algorithm 1.

For simplicity, we rewrite (12) as $\tilde{B}_k = \beta_k - B_{-k}$ where

$$\beta_k = \left(\sqrt{\frac{1}{4} + \frac{1}{2} \frac{\theta_k}{A_k}} - \frac{1}{2} \right)^2. \quad (13)$$

Intuitively, β_k is a parameter indicating participant's quality, which captures the mapping relationship from model preference and training cost to participant's batchsize strategy, e.g., a larger β_k means a higher participating enthusiasm (larger $\tilde{B}_k$).

Since participants' model preferences θ_k are continuous variables in general, the probability that two participants have exactly the same preference would be zero under a given distribution. Hence, we focus on the case that the set of participants $\mathcal{K} = \{1, 2, \dots, K\}$ are sorted in descending order by β_k without ties, i.e., $\beta_1 > \dots > \beta_K$. Best response functions in (11) imply that there are three types of participants when the game reaches equilibrium. Correspondingly, we define the *type-1* (S_1), *type-2* (S_2) and *type-3* (S_3) participants. Specifically, $S_1 = \{i \in \mathcal{K} | B_i' > B_i^{\max}\}$, $S_2 = \{i \in \mathcal{K} | 0 \leq B_i' \leq B_i^{\max}\}$ and $S_3 = \{i \in \mathcal{K} | B_i' < 0\}$. We can

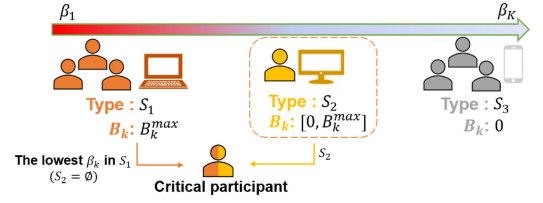


Fig. 2. The structure properties of Nash equilibrium in the COFL game.

show Theorem 1, which reveals the structural properties of Nash equilibrium in COFL.

Theorem 1. At the Nash equilibrium of the COFL game, the participants can be divided into the sets of type-1, type-2 and type-3. For any $i \in S_1, c \in S_2, j \in S_3$, we have $\beta_j < \beta_c < \beta_i$. Moreover, at most one participant belongs to type-2 participant.

The proof is given in Section B in Appendix A in the separate supplementary file, available online. As illustrated in Fig. 2, Theorem 1 indicates when the game reaches the Nash equilibrium (if it exists), S_1 , S_2 and S_3 are arranged in descending order of β_k . Among them, the type-2 participant is no more than one. As a result, we attempt to find a boundary between S_1 and S_3 called critical participant:

Definition 1. Participant c is a critical participant in COFL if and only if $\forall i \in \mathcal{K}, \beta_i < \beta_c, \tilde{B}_i = \beta_i - B_{-i} < 0$ and $\tilde{B}_c = \beta_c - B_{-c} \geq 0$ when the game reaches a Nash equilibrium (if it exists).

The critical participant belongs to type-1 and has the lowest β_k among type-1 participants if S_2 is a empty set. On the other hand, if participant c belongs to type-2, he must be the critical participant. That is, the critical participant is the boundary between S_1 and S_3 . At Nash equilibrium, participants with β_k higher than β_c belong to S_1 . Otherwise, participants belong to S_3 . Based on Theorem 1, the critical participant has a vital role in the Nash equilibrium, since others must belong to S_1 or S_3 . For the convenience of presentation, we predefine of participant c 's equilibrium structure in COFL game.

Definition 2 (participant's equilibrium structure). Given a participant $c \in \mathcal{K}$, (1) $\forall i \in \mathcal{K}, \beta_i > \beta_c$, we set $B_i = B_i^{\max}$. (2) $\forall j \in \mathcal{K}, \beta_j < \beta_c$, we set $B_j = 0$. We call the strategy profile $(B_1, \dots, B_{c-1}, B_{c+1}, \dots, B_K)$ as the participant c 's equilibrium structure, where each $B_i (i \neq c)$ satisfy (1) and (2).

We consider the conditions under which a participant c becomes the critical participant at equilibrium and derive the equivalent form of Nash equilibrium in Theorem 2.

Theorem 2. If there exists a participant c with equilibrium structure $(B_1, \dots, B_{c-1}, B_{c+1}, \dots, B_K)$, then one of the following properties holds for the COFL game:

- Property 1: $B_c^* = B_c^{\max}$ and $\beta_{c+1} < B_{nash} < \beta_c$.
- Property 2: $B_c^* = \tilde{B}_c$ and $B_{nash} = \beta_c$.

Where B_{nash} is the global batchsize (i.e., total batchsize contributed by the participants) at the equilibrium. In this case, Nash equilibrium exists in the game, and the participant c is critical participant. Also, if the COFL game reaches a Nash equilibrium (if it exists), one of the two properties is satisfied.

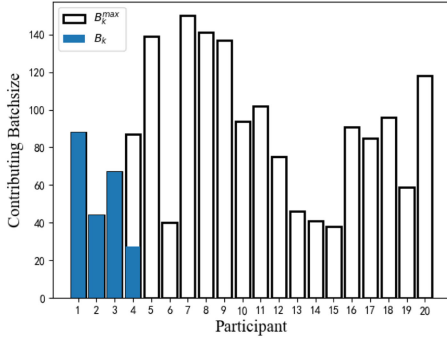


Fig. 3. The “free-riding” phenomenon in COFL.

The proof is given in Section C in Appendix A in the separate supplementary file, available online. In Theorem 2, Property 1 considers a case where critical participant c belongs to *type-1* and Property 2 corresponds to c as a *type-2* participant. Based on Theorem 2, we can search an equilibrium through considering the relationships between global batchsize and β_k . To account the global batchsize by all the participants, we define a global batchsize function $F_o(c, B_c^{\text{con}})$ in terms of the critical participant c and its batchsize B_c^{con} as

$$F_o(c, B_c^{\text{con}}) = \sum_{k=1}^{c-1} B_k^{\max} + B_c^{\text{con}} + \sum_{k=c+1}^K 0, \quad (14)$$

$$\text{s.t. } c \in \mathcal{K}, \quad (14a)$$

$$0 \leq B_c^{\text{con}} \leq B_c^{\max}. \quad (14b)$$

The first and third terms in the function (14) originates the batchsize contributions by *type-1* and *type-3* participants according to Theorem 2. For the feasibility of such algorithm design, we explore the existence and uniqueness of Nash equilibrium, which is guaranteed by Theorem 3.

Theorem 3. The contribution-oblivious FL game must admit a unique Nash equilibrium.

The proof is given in Section D in Appendix A in the separate supplementary file, available online. Based on above analysis, we introduce our approach on how to compute the Nash equilibrium in COFL in Algorithm 1. The key idea is to search a participant c satisfying Property 1 or 2 in Theorem 2. If we have found the critical index c and his strategy in Nash equilibrium, we can then know others' optimal strategies, i.e., $\beta_i > \beta_c, B_i = B_i^{\max}$ and $\beta_j < \beta_c, B_j = 0$. As shown in Algorithm 1, we search the lowest index i satisfying $F_o(i, B_i^{\max}) > \beta_i$. Here, $i - 1$ or i is the critical participant. Line 9 and 10 in Algorithm 1 calculate the critical participant's strategy when others' are fixed.

3.3 Free-Riding Phenomenon

When COFL reaches the Nash equilibrium, Theorem 1 shows that *type-1* participants contribute their maximum data batchsize in model training. However, the *type-3* participants can obtain the FL model without any cost, which raises the critical issue of free-riding. By executing Algorithm 1, Fig. 3 shows the free-riding phenomenon of a group, which is made up of 10 *High quality* and 10 *Low quality* participants (the setting of these two kinds of participant is detailed in Section 6). The critical participant is 4. As a result, the participants with β less

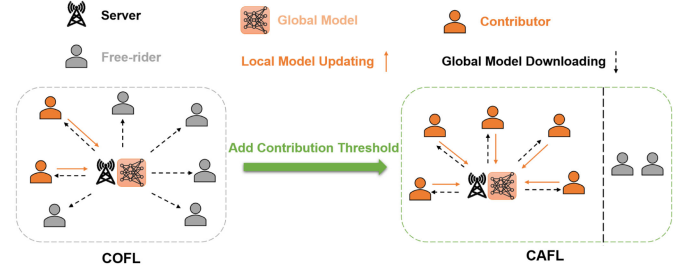


Fig. 4. The comparison of COFL and CAFL.

than β_4 (participants indexing from 5 to 20) choose to free-ride and contribute nothing. This illustrates that the free-riding problem by a significant portion of participants would greatly break the fairness and harm the motivations for collaboration in participant-centric FL if we directly apply COFL game model based on collaboration strategy.

Algorithm 1. COFL Nash Equilibrium Computing Algorithm

Input: K participants (descending order by β_k)
Output: Strategy profile $(B_1, B_2, \dots, B_K)$

- 1: Search lowest index i satisfying $F_o(i, B_i^{\max}) > \beta_i$ using binary search.
- 2: **if** $F_o(i, 0) \leq \beta_i$ **then**
- 3: i is the critical participant denoted as c .
- 4: **else**
- 5: $i - 1$ is the critical participant denoted as c .
- 6: **end if**
- 7: $\forall k \in \mathcal{K}, \beta_k > \beta_c$, we set $B_k = B_k^{\max}$.
- 8: $\forall k \in \mathcal{K}, \beta_k < \beta_c$, we set $B_k = 0$.
- 9: Calculate $\tilde{B}_c$ using (12).
- 10: Calculate B_c using (11).
- 11: **return** strategy profile $(B_1, B_2, \dots, B_K)$

4 CONTRIBUTION-AWARE FL GAME

In response to the challenges of free-riding phenomenon in COFL, we then propose a novel contribution-aware FL (CAFL) game model, where a participant will be excluded from FL if his contributed batchsize is lower than a given minimum threshold batchsize B_{th} . As illustrated in Fig. 4, when a participant is excluded, he cannot receive the trained FL model from the server. As a result, more participants are willing to contribute in order to obtain the FL model as the return. In the followings, we will formulate the CAFL game and focus on investigating the equilibrium solution.

4.1 Game Formulation and Best Response in CAFL

Different from COFL, we add a minimum threshold mechanism in order to mitigate the free-riding issue by imposing the minimum contribution requirement. The minimum threshold satisfies $0 < B_{th} \leq B_{th}^{\max} = \min_{k \in \mathcal{K}} (B_k^{\max})$. Given a B_{th} , the utility function of participant k is defined as

$$U_k(B_k, f_k) = \begin{cases} \theta_k \ln(1 + \sqrt{B_{-k} + B_k}) - \frac{1}{2} \varphi_k \alpha C_k B_k f_k^2 \\ -\gamma_k \frac{C_k B_k}{f_k}, & \text{if } B_k \in [B_{th}, B_k^{\max}]. \\ 0, & \text{if } B_k = 0. \end{cases} \quad (15)$$

Here, $U_k(0, f_k) = 0$ means that a participant contributes nothing and is excluded to participate in FL with zero utility. The discussion of strategy in interval $(0, B_{th})$ is meaningless, since the participant will be excluded from FL. We then define the CAFL game model as follows:

Game 2 (CAFL Game).

- **Players:** The set $\mathcal{K}$ of participants.
- **Strategies:** $B_k \in \{0\} \cup [B_{th}, B_k^{max}]$, $f_k \in [f_k^{min}, f_k^{max}]$.
- **Utilities:** The utility $U_k(B_k, f_k)$ in (15) for each $k \in \mathcal{K}$.

In CAFL, we assume that if participant k participates in CAFL and his utility $U_k \geq 0$, he is willing to participate in the FL. Accordingly, we derive the conditions for participants to participate in FL as follows:

Proposition 2. A participant k is willing to participate in federated learning in CAFL if and only if the following holds:

$$B_{-k} \geq \left(e^{\frac{A_k B_{th}}{\theta_k}} - 1 \right)^2 - B_{th}. \quad (16)$$

The proof is given in Section A in Appendix B in the separate supplementary file, available online. For simplicity, we define a function Φ to represent the right hand of (16) as

$$\Phi(B_{th}, \frac{\theta_k}{A_k}) = \left(e^{\frac{A_k B_{th}}{\theta_k}} - 1 \right)^2 - B_{th} \triangleq \Phi_k. \quad (17)$$

Proposition 2 shows that whether a participant participates in FL depends on B_{th} , participant's parameter θ_k and unit training cost A_k . From (16), given a B_{th} , a lower $\frac{\theta_k}{A_k}$ (which also indicates a smaller β_k as per (13)) implies a higher barrier Φ_k for a participant to benefit. In this case, only if others contribute sufficient data, participants with small β_k would participate in FL. Based on the above discussion, we formally derive participant's best response strategy as follows.

$$B_k^* = \begin{cases} B_k^{max}, & \text{if } B_k^{max} < \tilde{B}_k. \\ \tilde{B}_k, & \text{if } B_{th} \leq \tilde{B}_k \leq B_k^{max}. \\ B_{th}, & \text{if } \tilde{B}_k < B_{th} \text{ and } B_{-k} \geq \Phi_k. \\ 0, & \text{if } \tilde{B}_k < B_{th} \text{ and } B_{-k} < \Phi_k. \end{cases} \quad (18)$$

And, f_k^* and $\tilde{B}_k$ are obtained by (10) and (12) respectively.

Similar to COFL, we can define four types of participants as *type-1* (P_1), *type-2* (P_2), *type-3* (P_3) and *type-4* (P_4) participants in Nash equilibrium, which correspond to the first, second, third and fourth conditions in (18).

4.2 Equivalent Form of Nash Equilibrium

In this subsection, we focus on deriving the characteristics of Nash equilibrium (Theorem 4), based on which we give the equivalent form of Nash equilibrium (Theorem 5).

Similar to COFL, all participants have been sorted in descending order by β_k . Theorem 4 describes the characteristic of Nash equilibrium in CAFL.

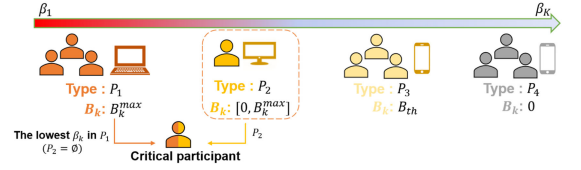


Fig. 5. The structure properties of Nash equilibrium in the CAFL game.

Theorem 4. If CAFL reaches a Nash equilibrium, the participants can be divided into type-1, type-2, type-3 and type-4. For any $i \in P_1, c \in P_2, j \in P_3, k \in P_4$, we have $\beta_k < \beta_j < \beta_c < \beta_i$. Moreover, at most one participant belongs to type-2 participant.

The proof is given in Section B in Appendix B in the separate supplementary file, available online. As illustrated in Fig. 5, Theorem 4 shows that when the CAFL game reaches the Nash equilibrium, P_1, P_2, P_3 and P_4 are arranged in descending order of β_k . Similarly, we describe the boundary between P_1 and P_3 and the critical participant and equilibrium structure are redefined as:

Definition 3. Participant c is a critical participant in CAFL game if and only if $\forall i \in \mathcal{K}, \beta_i < \beta_c, B_i = \beta_i - B_{-i} < B_{th}$ and $\tilde{B}_c = \beta_c - B_{-c} \geq B_{th}$ when the game reaches a Nash equilibrium (if it exists).

Definition 4 (participant's equilibrium structure). Given a participant $c \in \mathcal{K}$, (i) $\forall i \in \mathcal{K}, \beta_i > \beta_c, B_i = B_i^{max}$; (ii) $\forall j \in \mathcal{K}, \beta_j < \beta_c, B_{-j} \geq \Phi_j$, we set $B_j = B_{th}$; (iii) $\forall l \in \mathcal{K}, \beta_l < \beta_c, B_{-l} < \Phi_l$, we set $B_l = 0$. We call the strategy profile $(B_1, \dots, B_{c-1}, B_{c+1}, \dots, B_K)$ as the participants' equilibrium structure in the CAFL game, where $B_i (i \neq c)$ satisfy (i), (ii) and (iii).

Based on Theorem 4, we summarize the conditions of one participant becoming the critical participant in CAFL and derive equivalent form of Nash equilibrium in Theorem 5:

Theorem 5. If there exists a participant c with equilibrium structure, then the CAFL game satisfies one of three properties:

- **Property 3:** $B_c^* = B_c^{max}$, $\beta_{c+1} - B_{th} < B_{nash} < \beta_c$ and $B_{c+1} = 0$.
- **Property 4:** $B_c^* = B_c^{max}$, $\beta_{c+1} < B_{nash} < \beta_c$ and $B_{c+1} = B_{th}$.
- **Property 5:** $B_c^* \neq 0$ and $B_{nash} = \beta_c$.

Where B_{nash} is the global batchsize (i.e., total batchsize contributed by the participants) at the equilibrium. In this case, the Nash equilibrium exists in CAFL game, and the participant c is a critical participant. Also, if the CAFL game reaches a Nash equilibrium (if it exists), one of the three properties above is satisfied.

The proof is given in Section C in Appendix B in the separate supplementary file, available online. The equivalent form of Nash equilibrium motivates us to compute Nash equilibrium using the rules of (i), (ii) and (iii) in Definition 4 until the one of three properties holds. Theorem 5 shows the relation between global batchsize and participants' β_k at the equilibrium when c is critical participant. Accordingly, we define the global batchsize function at the equilibrium as

TABLE 1
Example Without Nash Equilibrium

	θ_k	A_k	β_k	B_k^{max}	B_{th}
Participant 1	103.41	1	45.00	100	20
Participant 2	9.39	1	2.97	100	20

$$F_c(c, B_c^{con}) = \sum_{k=1}^{c-1} B_k^{max} + B_c^{con} + \max_{\mathbb{B}} \left(\sum_{j=c+1}^K B_j \right), \quad (19)$$

$$\text{s.t. } c \in K, \quad (19a)$$

$$B_{th} \leq B_c^{con} \leq B_c^{max}, \quad (19b)$$

$$B_j = B_{th}, \text{ if } B_{-j} \geq \Phi_j, \quad (19c)$$

$$B_j = 0, \text{ if } B_{-j} < \Phi_j. \quad (19d)$$

Here, $\mathbb{B} = (B_{c+1}, \dots, B_K)$. The first term of (19) corresponds to rule (i). The second term is the critical participant c 's batchsize constrained by (19b). The third term is constrained by (19c) and (19d) originating from rules (ii) and (iii) respectively. Given a pair of c and B_c^{con} , the summation of third term of (19) may correspond to multiple values. To obtain the equilibrium with the largest batchsize, we choose the maximum global batchsize at the equilibrium as our solution. Since we need to consider the relation between global batchsize and β_k , for convenience, we express three properties in Theorem 5 in the equivalent forms using the global batchsize function F_c as follows:

- Property 3: $\beta_{c+1} - B_{th} < F_c(c, B_c^{max}) < \beta_c$ and $B_{c+1} = 0$.
- Property 4: $\beta_{c+1} < F_c(c, B_c^{max}) < \beta_c$ and $B_{c+1} = B_{th}$.
- Property 5: There exists a $B_c^{con} \in [B_{th}, B_c^{max}]$ such that $F_c(c, B_c^{con}) = \beta_c$.

Thus, we can search the equilibrium by checking above properties.

However, Nash equilibrium does not necessarily exist in CAFL game. Table 1 shows an example without Nash equilibrium. We obtain $F_c(1, B_{th}) = 20 < \beta_1 < F_c(1, B_1^{max}) = 100 + 20$.⁴ We calculate $\Phi(B_{th}, \frac{\theta_2}{A_2}) = 34.97$. That is, the participant 2 participates with $B_2 = 20$ in FL if and only if participant 1's batchsize is greater than 34.97. Obviously, we fail to find a B_1 satisfying $F_c(1, B_1) = \beta_1$, i.e., the Property 5 does not hold. Absence of Nash equilibrium indicates that we can not compute Nash equilibrium as the same as that in the COFL game. In what follows, we will first introduce a partial form of CAFL game, and based on which, we compute Nash equilibrium and address the CAFL game without Nash equilibrium.

4.3 Nash Equilibrium for Partial Form of CAFL Game

In this subsection, we first consider the case that the pair of c and B_c^{con} is given, and accordingly define a partial form of the CAFL game consisting of the participants in set of $\{c+1, \dots, K\}$. We prove that such a partial game must possess Nash equilibria.

4. Since $B_1 = B_1^{max} = 100$, the global batchsize satisfies $B > \beta_1$, which means Property 3 or Property 4 does not hold.

Specifically, when given a c and B_c^{con} , we focus on the participants with their β_k lower than β_c and consider a new partial game among them where participants only have two strategies, i.e., B_{th} and 0, which is due to the constraints (19c) (19d). To sum up, the partial form of CAFL game is defined as

$$G(H, \mathcal{J}, \mathcal{S}, U_j), \quad (20)$$

where

- External parameter: $H = \sum_{k=1}^{c-1} B_k^{max} + B_c^{con}$.
- Participant set: $\mathcal{J} = \{c+1, \dots, K\}$.
- Participant strategy: $\mathcal{S} = \{0, B_{th}\}$.
- Utility function: U_k is defined as

$$U_j(B_j, f_j^*) = \theta_j \ln(1 + \sqrt{H + B_{-j} + B_j}) - A_j B_j. \quad (21)$$

Here, $B_{-j} = \sum_{i \in \mathcal{J} \setminus \{j\}} B_i$ and $U_j(0, f_j^*) = 0$. Note that in the partial game G , we introduce the external parameter H indicating that participants in $\{1, \dots, c\}$ outside the partial game G that adopt the given equilibrium strategies of B_k^{max} and B_c^{con} . Similar to Proposition 2, the participants' optimal strategies in game G can be determined by the following condition:

$$B_{-j} \geq \Phi(B_{th}, \frac{\theta_j}{A_j}) - H. \quad (22)$$

According to the third term of the global batchsize function $F_c(c, B_c^{con})$ in (19), we would like to choose the equilibrium with the largest global batchsize $B_G = \sum_{i \in \mathcal{J}} B_i$ of the partial form game G . Our method to achieve Nash equilibrium with the largest global batchsize in G is given in Algorithm 2. The key idea of Algorithm 2 is to set all participants' strategies to B_{th} initially. And then, it traverses each participant in reverse order to find the first participant who satisfies the condition in (22).

Algorithm 2. Partial Game Nash Equilibrium Computation Algorithm

Input: $G(H, \mathcal{J}, \mathcal{S}, U_j)$

Output: Strategy profile $(B_{c+1}, B_{c+2}, \dots, B_K)$

- 1: Sort participants in ascending order by $\Phi_{c+1} < \dots < \Phi_K$.
- 2: Initially set $\forall j \in \mathcal{J}, B_j = B_{th}$.
- 3: **for** $j = K$ to $c+1$ **do**
- 4: **if** $B_{-j} < \Phi(B_{th}, \frac{\theta_j}{A_j}) - H$ **then**
- 5: $B_j = 0$.
- 6: **else**
- 7: **break**.
- 8: **end if**
- 9: **end for**
- 10: **return** $(B_{c+1}, B_{c+2}, \dots, B_K)$

Theorem 6. Partial game G must admit Nash equilibria. Algorithm 2 can achieve a Nash equilibrium with largest global batchsize with complexity of $O(|\mathcal{J}|)$.

The proof is given in Section D in Appendix B in the separate supplementary file, available online. Hence, in the rest of paper, we use the Algorithm 2 to determine the equilibrium

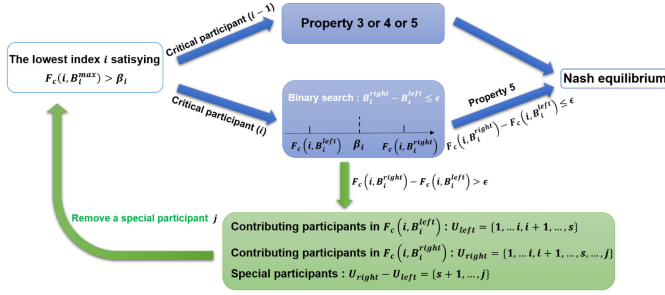


Fig. 6. The procedure of computing Nash equilibrium of CAFL game.

solution for the game G . Besides, given c and B_c^{con} , we can calculate the value of $F_c(c, B_c^{con})$ through Algorithm 2.

4.4 Nash Equilibrium for Complete Form of CAFL Game

We next consider the equilibrium solutions of the complete form of CAFL game by all the participants. According to Theorem 5, if we can search a critical participant c such that one of the properties in Theorem 5 is satisfied, then CAFL must have a Nash equilibrium. Otherwise, there would exist some *special participants*, who are not *type-3* and *type-4* but in between causing instable behaviors in the CAFL game, which would impede forming stable collaboration in participant-centric FL. We hence remove these special participants one by one until we can compute a Nash equilibrium.

The procedure of computing Nash equilibrium is illustrated in Fig. 6. Specifically, based on definition of the global batchsize function F_c in (19), $F_c(c, B_c^{max})$ is increasing with respect to c . We hence first search the lowest participant index i satisfying $F_c(i, B_i^{max}) > \beta_i$, which indicates $F_c(i-1, B_{i-1}^{max}) \leq \beta_{i-1}$. Then, the possible critical participant must be $i-1$ or i .

- $i-1$ as the critical participant: we then first determine whether $i-1$ is a critical participant by checking the relationship among $F_c(i-1, B_{i-1}^{max})$, β_{i-1} and β_i to verify whether Property 3 or 4 or 5 holds.
- i as the critical participant: Otherwise, when i is a critical participant, Property 3 or 4 does not hold, since $F_c(i, B_i^{max}) > \beta_i$. We then attempt to search a $B_i^{con} \in [B_{th}, B_i^{max}]$ satisfying $F_c(i, B_i^{con}) = \beta_i$ to verify whether Property 5 holds with i as the critical participant. Note that, $F_c(i, B_i^{con})$ is also increasing with respect to B_i^{con} , since larger first and second terms of F_c (larger H in game G) means larger third term. Thus, Property 5 (i as critical participant) indicates that $F_c(i, B_{th}) \leq \beta_i < F_c(i, B_i^{max})$. This motivates us to process *binary search* to finish searching. We predefine a small threshold as search accuracy ϵ . When the Property 5 holds (i as critical participant), there exists a $B_i^{con} \in [B_{th}, B_i^{max}]$ satisfying $F_c(i, B_i^{con}) = \beta_i$. We will obtain the result: $B_{right} - B_{left} \leq \epsilon$ and $F_c(i, B_{right}) - F_c(i, B_{left}) \leq \epsilon$. We can choose one of B_{left} and B_{right} as i 's strategy. Here, B_{left} and B_{right} are boundaries of the interval of binary search.
- *Nash equilibrium dose not exist*: when the Property 5 does not hold, we will fail in searching a $B_i^{con} \in [B_{th}, B_i^{max}]$ satisfying $F_c(i, B_i^{con}) = \beta_i$. In this case, the CAFL game does not have a Nash equilibrium. As a result, we will try to remove some special participants in the game.

such that the refined CAFL game has a Nash equilibrium. Intuitively, these special participants can be plausible contributing participant candidates but cannot satisfy Property 5 exactly. In this case, they are very sensitive to minor changes of other participants' decisions, causing the instable equilibrium behavior of switching between contributing ($B_i = B_{th}$) and quitting ($B_i = 0$) after the best response adjustments by others. Formally, the above special participants' characteristic will cause the *binary search* stopping with following results: $B_{right} - B_{left} \leq \epsilon$ and $F_c(i, B_{right}) - F_c(i, B_{left}) > \epsilon$, with $F_c(i, B_{left}) < \beta_i < F_c(i, B_{right})$. This indicates that $F_c(i, B_i^{con})$ is not a continuous function respect to B_i^{con} in the small interval $B_i^{con} \in [B_{left}, B_{right}]$. According to the global batchsize function $F_c(i, B_{left})$ in (19), we denote U_{left} as the contributing participant set in which all the participants' strategy batchsizes is non-zero. Similarly, the contributing participant set U_{right} corresponds to $F_c(i, B_{right})$. The set of special participants can be obtained by $U_{right} - U_{left}$. We can then remove a special participant with the lowest β_k and repeat searching the critical participant until we succeed in achieving Nash equilibrium.

Note that when (refined) CAFL game has a Nash equilibrium with the identified critical participant c , we can compute the equilibrium strategies as follows: 1) we set $B_k = B_k^{max}$, $\forall k \in \{1, \dots, c-1\}$; 2) $B_c = B_c^{con}$; 3) obtain the strategies for the participants in the set $\{c+1, \dots, K^{new}\}$ using Algorithm 2 based on the partial game G . Here, K^{new} denotes the total number of participants in the refined CAFL game after removing the special participants if the original CAFL game does not have a Nash equilibrium. For the removed participants, we have $B_k = 0$ and they are excluded from obtaining the trained FL model.

Algorithm 3. CAFL Nash Equilibrium Computing Algorithm

Input: K participants, batchsize threshold B_{th}

Output: $(B_1, B_2, \dots, B_K)$

- 1: Search the lowest participant's index i satisfying $F_c(i, B_i^{max}) > \beta_i$.
- 2: Check Property 3 or 4 or 5 ($i-1$ as the critical participant) through the relationship among $F_c(i-1, B_{i-1}^{max})$, β_{i-1} and β_i .
- 3: **if** $i-1$ is the critical participant **then**
- 4: Construct equilibrium strategies by setting $c = i-1$.
- 5: Goto Line 16.
- 6: **else**
- 7: Binary search a $B_i^{con} \in [B_{th}, B_i^{max}]$ satisfying $F_c(i, B_i^{con}) = \beta_i$.
- 8: **if** succeed in searching B_i^{con} **then**
- 9: Construct equilibrium strategies by setting $c = i$.
- 10: Goto Line 16.
- 11: **else**
- 12: Remove the special participant with highest index j .
- 13: Goto Line 1.
- 14: **end if**
- 15: **end if**
- 16: **return** equilibrium strategies $(B_1, B_2, \dots, B_K)$

Due to space limit, more analysis on the above equilibrium construction procedure is detailed in Section E in

Appendix B in the separate supplementary file, available online.

5 CAFL GAME BASED ALGORITHM DESIGN AND IMPLEMENTATION

In this section, we summarize the results above to form a CAFL game based collaboration strategy for participant-centric FL. We further discuss the implementation and potential problems for the proposed mechanism in realistic deployment.

5.1 Algorithm Design

We first propose our approach to compute a Nash equilibrium for the CAFL game in Algorithm 3, with the key idea of searching the correct critical participant. In Line 1, we first search the critical participant ($i-1$ or i) using the binary search. In Lines 2 and 3, if $i-1$ is the critical participant at equilibrium, his strategy must be B_{i-1}^{max} . In Lines 6 and 7, we consider the case that i is the critical participant. Line 8 corresponds to Property 5 (i as critical participant). In Line 12, when Property 5 does not hold, we fail in achieving Nash equilibrium. Only one participant with the lowest β_k in the special participants (obtained in binary search) is removed. In Line 13, we repeat all steps and go to Line 1. We can characterize the convergence of Algorithm 3 in Theorem 7.

Theorem 7. In Algorithm 3, the maximum number of iterations (i.e., executions of line 13) is $K - i^*$, where i^* is the lowest index i satisfying $F_c(i, B_i^{max}) > \beta_i$ in the original CAFL game. The solution searched by Algorithm 3 is a Nash equilibrium for the refined CAFL game of the set of the remaining participants.

The proof is given in Section F in Appendix B in the separate supplementary file, available online. We further analyze the complexity of Algorithm 3. We first obtain the complexity for one iteration (Line 1-12). The complexity of calculating $F_c(i, B_i^{max})$ is $O(K)$. Hence, the complexity of Line 1 is $O(K \log(K))$. In Line 4, constructing equilibrium strategy for each participant, the complexity is $O(K)$. In Line 7, the complexity of binary search is $O(\log(\frac{B^{max}-B_{th}}{\epsilon}))$. Here, $B^{max} = \max_{k \in K} (B_k^{max})$. Thus, the complexity of one iteration is $O(K \log(\frac{B^{max}-B_{th}}{\epsilon}) + K \log(K))$. Based on Theorem 7

(the maximum number of iteration is bounded by K), the complexity of Algorithm 3 is $O(K^2 \log(\frac{B^{max}-B_{th}}{\epsilon}) + K^2 \log(K))$. Note that, a lower ϵ leads to a higher complexity.

We next consider boosting the global efficiency of the CAFL game by searching the optimal threshold batchsize $B_{th} \in [1, B_{th}^{max}]$ to maximize the total utility. Specifically, we define the total utility of all in equilibrium defined as

$$TU = \sum_{k \in \mathcal{U}} (\theta_k \ln(1 + \sqrt{B_{-k} + B_k}) - A_k B_k). \quad (23)$$

Here, $\mathcal{U}$ represents the contributing participant set in which all the participants' strategy batchsizes is non-zero in Nash equilibrium. For practical implementation, we can only consider the choices of discrete minimum threshold batchsize, i.e., $B_{th} = 1, 2, \dots, B_{th}^{max}$. Given a B_{th} , the total utility can be

TABLE 2
The Average Running time(s) of Optimal Threshold Searching

B_{th}^{max}	100	500	1000
K			
100	0.5193	2.270	4.374
500	3.185	15.22	28.52
1000	7.080	33.98	55.31

calculated by performing Algorithm 3 to obtain the equilibrium solution. Thus, we can obtain the optimal threshold batchsize through exhaustive search. The running time for searching the optimal threshold batchsize is illustrated in Table 2. B_{th}^{max} indicates the number of execution of Algorithm 3. As for average execution time of Algorithm 3, for example, given a fixed $B_{th}^{max} = 100$ and $K = 100$, the average execution time of Algorithm 3 (determine a Nash equilibrium points) is $\frac{0.5193}{B_{th}^{max}} = 0.05193$ s, which indicates the high computational efficiency of Algorithm 3. Note that running time for searching the optimal threshold batchsize is short in general, the average running time (repeated over 20 times) is less than 56 seconds when the number of participants $K = 1000$ and the maximum batchsize $B_{th}^{max} = 1000$, which is negligible compared with the time consuming FL model training process.

5.2 Discussion on Implementation and Potential Problems

In practice, for deployment of the CAFL game based incentive mechanism, each participant can first calculate his unit training cost A_k locally based on its own information, and then reports the parameters of unit cost A_k , maximum data size B_k^{max} and model valuation θ_k to the FL server (which plays a neutral role in participant-centric FL). Based on the participants' reported information, the server will compute the optimal equilibrium strategies with optimal threshold for the participant-centric FL using the algorithms above, and then announce the strategies to the participants. We should emphasize that each participant has incentive to follow the announced strategy due to the property of Nash equilibrium and the enforcement of contribution threshold mechanism by the FL server.

We have achieved a lightweight mechanism to collaborate participants to finish the FL task. In what follows, we further discuss the potential problems in realistic deployment.

Model Security. Similar to many existing studies on incentive issues [13], [20], [25], we assume that all participants are selfish but have no intention of sabotaging the FL model. In terms of potential model security issues, using validation dataset is a lightweight and economic method to check quality of the uploaded model from each participant in order to defend against data poisoning attack or model manipulation (e.g., dirty-label data [21]). In response to the backdoor attack [26], we can refer to three state-of-art defense mechanisms to defend it from the perspective of certified defense [27], validation datasets detection [28] and robust aggregation [29], respectively. These

aforementioned mechanisms can be easily applied in our participant-centric FL scenario.

Non-IID Data. we assume that all participants have the common interest and would like to obtain the global trained FL model and their possess IID data, while neglecting the difference of data distribution among participants to some extend (non-IID). From economic perspective, non-iid case will not change the essential characteristic of COFL (free-riding phenomenon). The non-iid issue will be considered in future work. Intuitively, one potential extension is to adjust the participants' overall model valuation parameters, since the overall non-iid level in a large-scale group is relatively static.

Communication or Computation Disruption. we focus on the participants' collaboration in FL from theoretical perspective in this paper. In realistic deployment, a participant may experience failures in terms of communication or computation disruptions probabilistically. To address such a risk-aware FL collaboration scenario, one study direction is to integrate the prospect theory [30], [31] with our game model to analyze risk-aware decision making behaviors when facing with uncertainty. Intuitively, risk-aware decision makings would make the participants to be more conservative in collaboration. The rigorous analysis on this based on prospect theory is mathematically involved and out of scope of this study. We will consider it in a future work.

Truthful Parameter Reporting. the issue of participant's truthful parameter reporting is not the focus of this work and will be considered in a future work. Intuitively, if a participant exaggerates his model valuation, he would be required to contribute more. While, if he understates his valuation excessively, he would risk at getting removed from FL. In practice, as per the prior statistical distributions of participants' valuations, we can globally finetune the contribution threshold to balance the truthfulness and optimality.

6 NUMERICAL RESULTS

In this section, we evaluate CAFL performance on global batchsize and total utility in different structures of groups. Experiment results show that CAFL effectively alleviates the "free-riding" phenomenon. We also study the impact of parameters on participants' behaviors. We find that participant's position in group (i.e., the relative size of β_k) has a great influence on the participant's strategy. We will also evaluate the performance of the FL model achieved by CAFL and COFL in the realistic MNIST, FashionMNIST and CIFAR10 datasets.

6.1 Contribution-Aware FL Performance

Simulation Setting. We first divide each group into two types of participants. (i) *High quality (Hq):* The parameter θ_k varies from 50 to 100. The f_k^{min} and f_k^{max} are 0.3 GHz and 1.5 GHz. We set parameters $\varphi_k \in [1, 10]$ and $\gamma_k \in [10, 100]$. And, C_k is uniformly distributed in $[1.22 \times 10^6, 2.44 \times 10^6]$; (ii) *Low quality (Lq):* The parameter θ_k ranges from 0 to 10. We set parameters $\varphi_k \in [1, 20]$ and $\gamma_k \in [10, 200]$. Other parameter settings are the same as that of *High quality*. Parts of the above parameter settings are based on [19]. The device for simulation in this paper is equipped with a 8-core Intel(R) Core(TM) i7-8650 U CPU and 1 NVIDIA GeForce GTX 1060

TABLE 3
The Average Number of Contributing Participants in Difference Groups

Hq(%) K	0	25	50	75	100
20 (COFL)	1.13	2.66	3.41	3.86	4.26
20 (CAFL)	18.84	15.02	17.02	17.27	20
50 (COFL)	1.38	3.71	4.52	4.96	5.27
50 (CAFL)	45.74	43.28	45.22	47.69	50
100 (COFL)	1.53	4.57	5.28	5.89	6.30
100 (CAFL)	93.47	87.49	91.19	96.05	100

GPU. The device for model training (Section 6.1.2) is based on Ubuntu 18.04.05, CUDA v11.6 and Intel(R) Xeon(R) CPU (E5-2678 v3).

6.1.1 The Performance of CAFL versus COFL

In this subsection, we study the performance of CAFL in different structures of participant groups.

We study the results for different groups with different proportions of *High quality* participant under different numbers of participants. Here, $B_k^{max} \in [30, 150]$. In order to reduce the experimental error caused by parameters' randomness, we run 100 times under each type of groups and average the results. Note that, B_{th} in CAFL is the group optimal minimum threshold batchsize.

As shown in Table 3, in the COFL, under the same ratio of *High quality (Hq)* participant is unchanged, the proportion of non-contributing participants gradually increases with the number of participants in group. This shows that the larger of participant size in the, the more severe free-riding phenomenon. Besides, there are much more contributing participants in CAFL (more than 90% participants contributed in most cases) than in COFL, which means CAFL effectively alleviates the free-riding phenomenon. Furthermore, the number of contributing participants in CAFL with 25% *High quality* is less than that in CAFL with 0% *High quality*. The reason is as follows: When the proportion of *High quality* is 0%, the threshold B_{th} ranges from 9 to 12 when number of participants K equals to 20, 50 and 100, respectively. However, when the proportion of *High quality* is 25%, B_{th} ranges from 30 to 35. The high threshold batchsize B_{th} makes more low-quality participants choose to exit. When the proportion of *High quality* continues to increase from 25%, the threshold B_{th} does not increase much, leading the increasing number of contributed participants.

Figs. 7 and 8 show the growth rate of CAFL relative to COFL on the global batchsize and total utility, respectively. As the number of participants increases, the free-riding phenomenon becomes more and more serious, and hence performance gain (both on global batchsize and total utility) of CAFL is more superior. For instance, CAFL can achieve up to 9x growth in terms of the global data batchsize over COFL, which also implies a significant improvement on the FL model accuracy. Moreover, the growth rates of CAFL in both utilities and global batchsize over COFL increase with the number of total participants K , which demonstrate that CAFL can be more efficient for large-scale FL applications.

Note that for a fixed number of participants K , with a larger

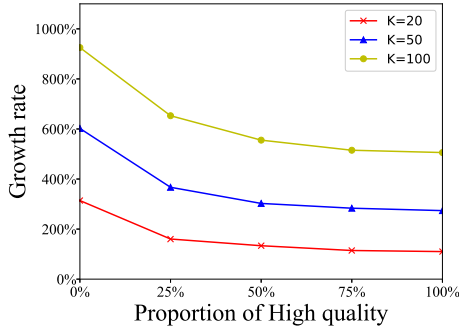


Fig. 7. Growth rate of CAFL on the global batchsize.

proportion of *High quality* participants, the impact of free-riding is slightly weakened in COFL, and hence the growth rates of CAFL change smoothly in Figs. 7 and 8.

6.1.2 The Training Performance of CAFL versus COFL

We further evaluate the performance of CAFL in MNIST and CIFAR10 dataset, compared with COFL.

MNIST and FashionMNIST (FMNIST) Setting. The standard MNIST [32] and FMNIST [33] consist of 60,000 training samples and 10,000 test samples. For MNIST setting, we use a multi-layer perception (MLP) network only with one hidden layer (256 hidden unit). For FMNIST setting, we utilize a network with 2 convolutional layers and 1 fully connected layer. We set learning rate $\eta = 0.01$ and local epoch equaling to 1. We conduct the experiment on three groups with different number of participants (50% *Hq* and 50% *Lq* participants) in COFL and CAFL respectively. Here, $B_k^{max} \in [200, 300]$.

CIFAR10 Setting. CIFAR10 dataset [34] has 50,000 training examples and 10,000 test examples. We use LeNet consisting of two sets of convolution and pooling layers, then two fully-connected layers with ReLU activation. The learning rate and local epoch are set to $\eta = 0.01$ and 1 respectively. The participant groups settings are the same as that in *MNIST setting*.

We compare the accuracy and loss function with COFL and CAFL in Figs. 9, 10, 11, 12, 13 and 14. In a finite number of global iterations, the training performance of CAFL is superior of COFL on three datasets, since CAFL can achieves a larger global batchsize FL. In terms of test accuracy ($T = 100$), CAFL are 43.26%, 48.68% and 55.71% more than COFL on MNIST dataset in three groups ($K = 20$, $K = 50$ and $K = 100$ respectively). On FMNIST dataset, the accuracy of CALF are 27.75%, 28.07%, and 29.47% more than

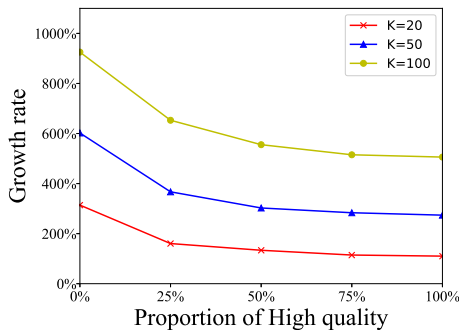


Fig. 8. Growth rate of CAFL on the total utility.

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on July 15, 2024 at 03:04:46 UTC from IEEE Xplore. Restrictions apply.

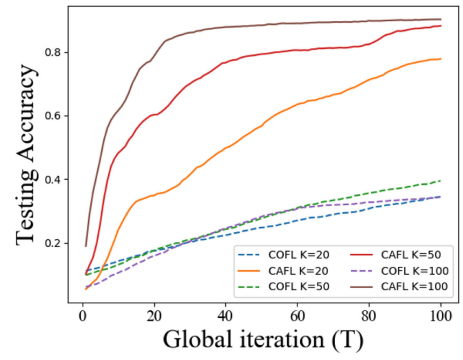


Fig. 9. The comparison of accuracy with COFL and CAFL on MNIST dataset.

that of COFL in three groups ($K = 20$, $K = 50$ and $K = 100$ respectively). Similarly, on CIFAR10 dataset, the accuracy of CAFL ($T = 100$) are 35.61%, 37.65% and 42.02% more than that of COFL in three groups ($K = 100$, $K = 200$ and $K = 300$ respectively). This implies that CAFL can greatly improve the training performance with a large number of total participants K .

6.1.3 The Performance of CAFL Versus Others Schemes

We compare with different schemes to study the superiority of the performance of CAFL. The details of each scheme can be listed as follows: *Uniform contribution*: Each of participant contributes the same amount of batchsize to maximize the total utility. *Optimal total utility*: Maximize the utilities of all the users regardless of individual rationality and free-riding. *Independent training*: Each of participant trains local model independently. For experiment settings, each of group consists of 10% *Hq* and 90% *Lq* participants and run 100 times under each type of group. We average performance on global batchsize, total utility and contributing participants relative to the number of participants for each scheme (Figs. 15, 16, and 17).

In Figs. 15 and 16, the CAFL is superior of *Uniform contribution* and *Independent training* both in the global batchsize and total utility. Even though *Uniform contribution* can alleviate free-riding problem, it inevitably stifles the high quality participants' motivation on training. In terms of total utility (Fig. 16), the performance of CAFL is within 92% of *Optimal total utility*. As number of participants increase, the

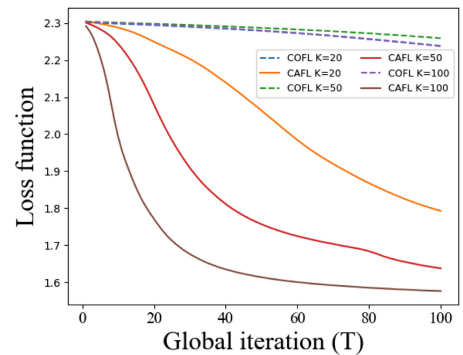


Fig. 10. The comparison of loss function with COFL and CAFL on MNIST dataset.

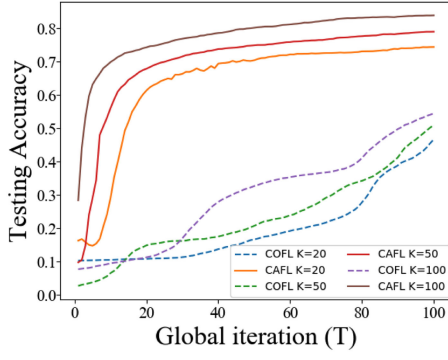


Fig. 11. The comparison of accuracy with COFL and CAFL on FMINIST dataset.

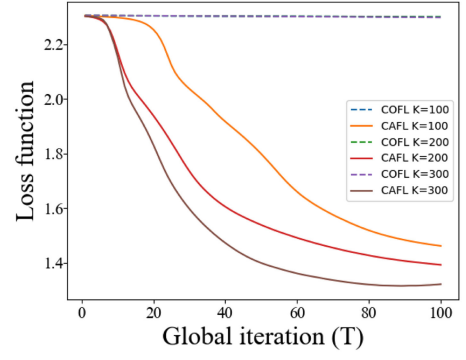


Fig. 14. The comparison of loss function with COFL and CAFL on CIFAR10 dataset.

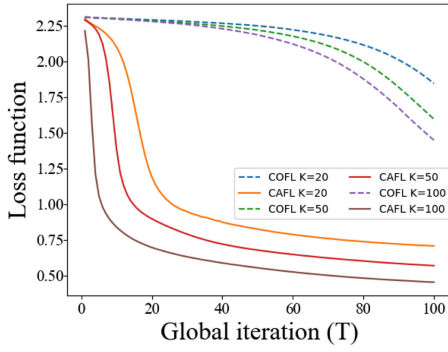


Fig. 12. The comparison of loss function with COFL and CAFL on FMINIST dataset.

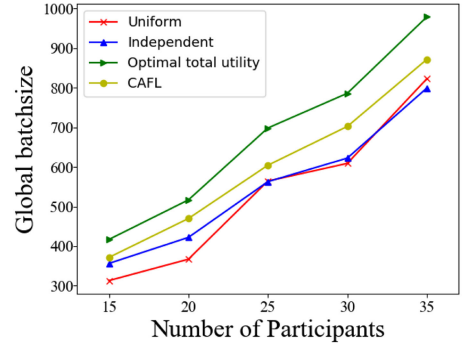


Fig. 15. The average global batchsize for each scheme.

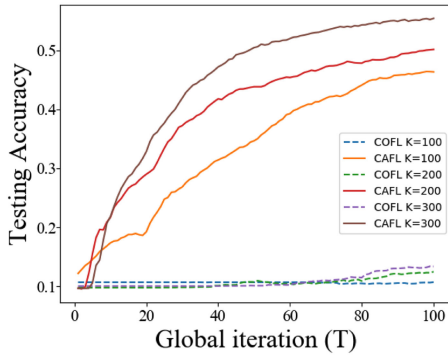


Fig. 13. The comparison of accuracy with COFL and CAFL on CIFAR10 dataset.

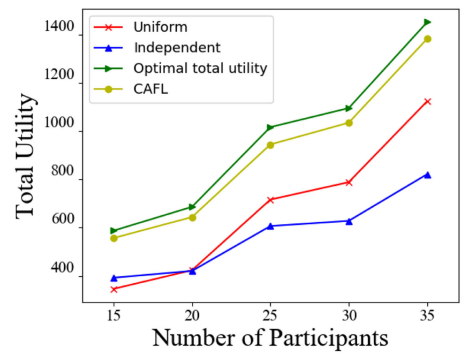


Fig. 16. The average total utility for each scheme.

difference between *Uniform contribution* and *Optimal total utility* becomes larger and larger, while the difference between CAFL and *Optimal total utility* is stable. Obviously, CAFL can be more efficient in large-scale FL application.

In Fig. 17, the contributing participants of *Optimal total utility* is the lowest. This is because it selects a part of high quality participants to serve all participants to achieve the highest total utility. *Independent training* is the highest one (as highest baseline), since each participant has motivation on training the local model ($\beta_k > 0$). By observing Fig. 17, the performance of CAFL is slightly lower than that of *Uniform contribution*, since CAFL removes a part of the lower quality participant to hold a result with higher total utility. It can be seen that CAFL is a good trade-off between total utility and contributing participants.

6.2 Participants' Behaviors

In this subsection, we study the impact of parameters on participants' strategies and analyze participants' behavior in CAFL.

For the intuitiveness of the experimental results, we use a group, which includes 10 *High quality* and 10 *Low quality* participants. Hence, all participants' B_k^{max} in group range to [30, 150]. In the following experiments, we modify the parameters of the 20th participant (current participant) with the lowest β_k in the group.

We first consider a CAFL with $B_{th} = 10$. Current participant's θ_k is uniformly distributed in [80, 160]. We set $\varphi_k = 10$ and $\gamma = 50$. The maximum batchsize of the current participant B_k^{max} is 300. Other parameters of the participant are the same as those in the group. Fig. 18 shows the relation between the strategy and θ_k . Intuitively, a larger θ_k means

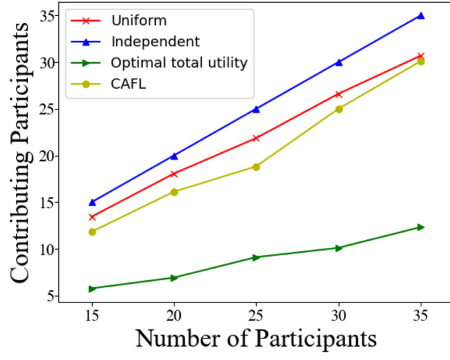


Fig. 17. The average contributing participants for each scheme.

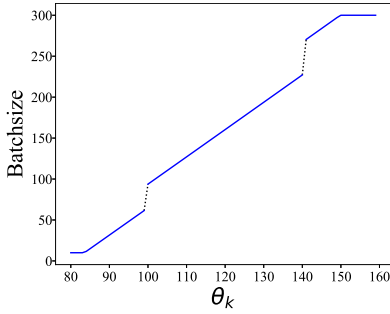


Fig. 18. Impact of θ_k on participants' strategies B_k .

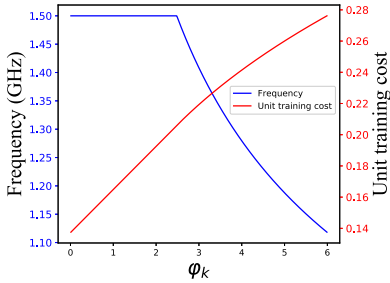


Fig. 19. Impact of φ_k on f_k .

higher preference about FL model, which indicates participants are willing to participate in the training with a large batchsize. We observe that the value of batchsize has some big jumps (dotted line) occasionally with the increase of θ_k , since relative sizes of current participants' β_k and others' have changed, which reflects participant's position in a community influences his strategy dramatically. Due to the characteristic of public goods, participants with higher preference about model will try their best to contribute more training data, which means these lower preference participants' needs are satisfied to some extent. Thus, these lower preference participants significantly decrease their contributions, which urges the current participant to dramatically increase the training data to meet his high demand.

In Figs. 19 and 20, φ_k ranges from 0.01 to 6. The parameter θ_k equals to 150. $\gamma_k = 167.56$ is identical to that in the group. B_{th} and B_k^{max} are 10 and 300 respectively. In Fig. 19, as φ_k increases, the participant is more concerned about energy consumption and attempts to decrease CPU frequency to minimize unit training cost. Fig. 20 shows that the batchsize does not decrease linearly, since participant

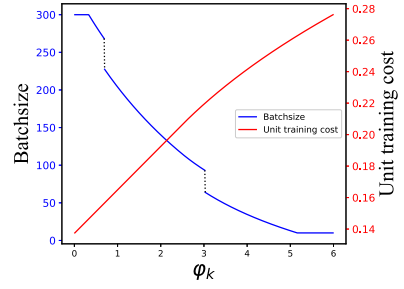


Fig. 20. Impact of φ_k on B_k .

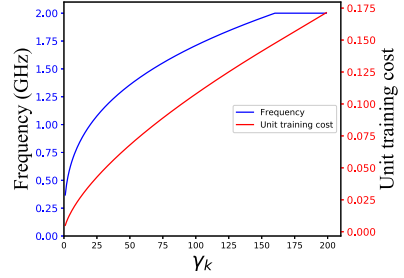


Fig. 21. Impact of γ_k on A_k .

adjusts frequency to slow down unit training cost growth. Besides, B_k plummets occasionally with the increase of unit training cost because of the change of the relative relationship in group.

We set $\varphi_k = 1$ to show the impact of γ_k . γ_k is uniformly distributed in $[1, 200]$. Here, $f_k^{max} = 2$ GHz. Fig. 21 illustrates that the change rates of frequency and unit training cost respect to γ_k are both numerically lower than those respect to φ_k . In reality, participants are more sensitive to energy consumption than training latency.

7 RELATED WORK

Google proposes federated learning framework using the federated average algorithm [6]. This algorithm aims to aggregate model parameters or gradients from mobile devices without revealing their raw data. Summarizing concept and applications in federated learning, Yang *et al.* classify federated learning into three types: horizontal federated learning, vertical federated learning and federated transfer learning [35].

FL Performance and Resource Optimization. Tran *et al.* consider the trade-off among model performance, latency and energy consumption in FL [19]. To accelerate model training, Ren *et al.* optimize allocation of communication resources and selection of local batchsize during training model [18]. Li *et al.* discuss the convergence of FL algorithm on non-iid data [36]. Khan *et al.* first propose self-organized FL and discuss an optimization problem of global federated learning time without a centralized server [37].

Incentive Mechanism in FL. Most of studies are based on an assumption that all participants participate in FL unconditionally. To attract more participants in federated learning, a well-designed incentive mechanism is necessary. Kang *et al.* apply contract theory to design an incentive mechanism to attract participants with high-quality data, i.e., high-quality data owners can receive more rewards [7]. Kang *et al.* combine reputation and contract theory to design a novel incentive

mechanism to ensure reliable FL [11]. Zhan *et al.* design a deep reinforcement learning-based incentive mechanism to obtain the optimal strategies of central server and participating edge nodes [12]. Sarikaya *et al.* considers the trade-off between training latency and payment for workers from the perspective of central server using stackelberg game [9]. Pandey *et al.* proposes a novel crowdsourcing framework to attract participant clients to provide a local model with a certain accuracy [8]. Existing papers are mainly concerned about a centralized task publisher scenario where there is only one model owner attracting others to complete FL. Different from previous studies, we discuss a self-organized FL where independent participants organize a community to collaborate on building a shared model in order to ensure a stable and fair federated learning system.

Game Theory. Game theory is a powerful tool to analyze the situation where many participants make optimal decisions considering effect from others' strategies. It has been successfully applied in data acquisition [24], data privacy preservation [38], incentive mechanism and resources optimization in FL [9], [39]. There are some common approaches to achieve a Nash equilibrium such as decision trees [38], best response dynamics, solving decision makers' best-response functions simultaneously [40].

8 CONCLUSION

In this paper, we develop a comprehensive theoretical framework for analyzing participants' behaviors in participant-centric federated learning. We propose the COFL game model and achieve the Nash equilibrium. To alleviate the free-riding phenomenon in COFL, we propose CAFL game model and establish a minimum threshold mechanism, which achieves the desired advantages of incentive-driven collaboration, free-riding mitigation and global efficiency boosting for participant-centric FL. We further show that optimal contribution threshold based CAFL game solution can significantly boost the amount of participation and system performance.

For the future work, we are going to study the implementation issue with truthful information reporting by globally fine-tuning the contribution threshold to balance the truthfulness and optimality. We will further address the issues of model security and non-IID data for participant-centric FL.

REFERENCES

- [1] S. Khare and M. Totaro, "Big Data in IoT," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol.*, 2019, pp. 1–7.
- [2] P. Yadav and S. Vishwakarma, "Application of Internet of Things and Big Data towards a smart city," in *Proc. 3rd Int. Conf. Internet Things, Smart Innov. Usages*, 2018, pp. 1–5.
- [3] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.
- [4] J. Xie *et al.*, "A survey on machine learning-based mobile Big Data analysis: Challenges and applications," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–19, 2018.
- [5] N. Rastogi, S. K. Singh, and P. K. Singh, "Privacy and security issues in Big Data: Through indian prospective," in *Proc. 3rd Int. Conf. On Internet Things: Smart Innov. Usages*, 2018, pp. 1–11.
- [6] 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.
- [7] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *Proc. IEEE VTS Asia Pacific Wireless Commun. Symp.*, 2019, pp. 1–5.
- [8] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, Z. Han, and C. S. Hong, "Incentivize to build: A crowdsourcing framework for federated learning," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.
- [9] Y. Sarikaya and O. Erceetin, "Motivating workers in federated learning: A Stackelberg game perspective," *IEEE Netw. Lett.*, vol. 2, no. 1, pp. 23–27, Mar. 2020.
- [10] 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.
- [11] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.
- [12] 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.
- [13] T. H. T. Le *et al.*, "An incentive mechanism for federated learning in wireless cellular network: An auction approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4874–4887, Aug. 2021.
- [14] 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.*, vol. 10, no. 2, pp. 1035–1044, Apr.–Jun. 2022.
- [15] Y. Hu, D. Niu, J. Yang, and S. Zhou, "FDML: A collaborative machine learning framework for distributed features," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2232–2240.
- [16] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "FedHome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2818–2832, Aug. 2022.
- [17] P. Kairouz *et al.*, "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*.
- [18] J. Ren, G. Yu, and G. Ding, "Accelerating DNN training in wireless federated edge learning system," 2019, *arXiv:1905.09712*.
- [19] 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.
- [20] 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.
- [21] L. Zhang, T. Zhu, P. Xiong, W. Zhou, and P. Yu, "A robust game-theoretical federated learning framework with joint differential privacy," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 04, 2022, doi: [10.1109/TKDE.2021.3140131](https://doi.org/10.1109/TKDE.2021.3140131).
- [22] R. Hu and Y. Gong, "Trading data for learning: Incentive mechanism for on-device federated learning," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.
- [23] 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.
- [24] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *Proc. IEEE Conf. Comput. Commun.*, 2012, pp. 1701–1709.
- [25] M. Tang and V. W. Wong, "An incentive mechanism for cross-silo federated learning: A public goods perspective," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
- [26] H. Wang *et al.*, "Attack of the tails: Yes, you really can backdoor federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 16070–16084.
- [27] C. Xie, M. Chen, P.-Y. Chen, and B. Li, "CRFL: Certifiably robust federated learning against backdoor attacks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11372–11382.
- [28] S. Andreina, G. A. Marson, H. Möllering, and G. Karame, "BaFFLe: Backdoor detection via feedback-based federated learning," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst.*, 2021, pp. 852–863.
- [29] C. P. Wan and Q. Chen, "Robust federated learning with attack-adaptive aggregation," 2021, *arXiv:2102.05257*.
- [30] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," in *Handbook of the Fundamentals of Financial Decision Making: Part I*. Singapore: World Scientific, 2013, pp. 99–127.

- [31] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1405–1418, Jun. 2020.
- [32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [33] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
- [34] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009. [Online]. Available: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>
- [35] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.
- [36] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on Non-IID data," 2019, *arXiv:1907.02189*.
- [37] L. U. Khan, M. Alsenwi, Z. Han, and C. S. Hong, "Self organizing federated learning over wireless networks: A socially aware clustering approach," in *Proc. Int. Conf. Inf. Netw.*, 2020, pp. 453–458.
- [38] L. Xu, C. Jiang, J. Wang, Y. Ren, J. Yuan, and M. Guizani, "Game theoretic data privacy preservation: Equilibrium and pricing," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 7071–7076.
- [39] L. U. Khan *et al.*, "Federated learning for edge networks: Resource optimization and incentive mechanism," 2019, *arXiv:1911.05642*.
- [40] W. N. James, *Game Theory*. Berlin, Germany: Springer, 2006.



Guangjing Huang received the BS degree from the School of Computer Science, South China Normal University, Guangzhou, China, in 2019, and the ME degree from the School of Computer Science and Engineering, Sun Yat-sen University (SYSU), Guangzhou, China, in 2021. He is currently working toward the PhD degree with the School of Computer Science and Engineering, SYSU. His research interests include federated learning, game theory, and mechanism design.



Xu Chen received the PhD degree in information engineering from The Chinese University of Hong Kong, in 2012, and worked as a postdoctoral research associate with Arizona State University, Tempe, from 2012 to 2014, and a Humboldt scholar fellow with the Institute of Computer Science of the University of Goettingen, Germany from 2014 to 2016. He is currently a full professor with Sun Yat-sen University, Guangzhou, China, director of Institute of Advanced Networking and Computing Systems, and the vice director of National Engineering

Research Laboratory of Digital Homes. 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, 2020 IEEE Computer Society best paper awards Runner-Up, 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 on Wireless Communications*, *IEEE Transactions on Vehicular Technology*, *IEEE Internet of Things Journal* and *IEEE Journal on Selected Areas in Communications (JSAC)* Series on Network Softwareization and Enablers.



Tao Ouyang received the BS degree from the School of Information Science and Technology, University of International Relations, Beijing, China, in 2017, and the ME degree from the School of Computer Science and Engineering, Sun Yat-sen University (SYSU), Guangzhou, China, in 2019. He is currently working toward the PhD degree with the School of Computer Science and Engineering, SYSU. His research interests include mobile edge computing, online learning, and optimization.



Qian Ma received the BS degree from the Beijing University of Posts and Telecommunications, China, in 2012, and the PhD degree from the Department of Information Engineering, The Chinese University of Hong Kong, in 2017. She is currently an associate professor of 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 include 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.



Lin Chen received the BE degree in radio engineering from Southeast University, China, in 2002, the MS degree in networking from the University of Paris 6, and the engineer and PhD degrees in computer science from Telecom Paris-Tech (ENST), in 2005 and 2008, respectively. He is currently working as a professor with the School of Computer Science and Engineering, Sun Yat-sen University. His main research interests include modeling and control for wireless networks, algorithm design, and analysis in networked systems.



Junshan Zhang (Fellow, IEEE) received the PhD degree from the School of ECE, Purdue University, in August 2000, and was on the faculty of the School of ECE, Arizona State University from 2000 to 2021. He is currently a professor with the ECE Department, University of California Davis. His research interests include information networks and data science, including edge intelligence, reinforcement learning, continual learning, network optimization and control, game theory, with applications in connected and automated vehicles, 5G and beyond, wireless networks, IoT data privacy/security, and smart grid. He is a recipient of the ONR Young Investigator Award, in 2005 and the NSF CAREER award, in 2003. He received the IEEE Wireless Communication Technical Committee Recognition Award, in 2016. His papers have won a few awards, including the Best Student paper with WiOPT 2018, the Kenneth C. Sevcik Outstanding Student Paper Award of ACM SIGMETRICS/IFIP Performance 2016, the Best Paper Runner-up Award of IEEE INFOCOM 2009 and IEEE INFOCOM 2014, and the Best Paper Award with IEEE ICC 2008 and ICC 2017. Building on his research findings, he co-founded Smartiply Inc, in 2015, a edge Computing startup company delivering boosted network connectivity and embedded artificial intelligence for IoT applications. He is currently serving as editor-in-chief for *IEEE Transactions on Wireless Communication* and an editor-at-large for *IEEE/ACM Transactions on Networking*. He was TPC co-chair for a number of major conferences in communication networks, including IEEE INFOCOM 2012 and ACM MOBIHOC 2015. He was the general chair for ACM/IEEE SEC 2017, WiOPT 2016, and IEEE Communication Theory Workshop 2007. He was a distinguished lecturer of the IEEE Communications Society. He was an editor for the *Computer Network Journal*, and an editor *IEEE Wireless Communication Magazine*.

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