Incentivize Contribution and Learn Parameters Too: Federated Learning with Strategic Data Owners

Drashthi Doshi[1], Aditya Vema Reddy Kesari[1], Avishek Ghosh[1], Swaprava Nath[1], Suhas S Kowshik[2], and Faizan Ahemad[2]

[1]IIT Bombay
[2]Amazon India

{drashthi,22b3985,avishek,swaprava}@iitb.ac.in, {kowssuhp,ahemf}@amazon.com

Abstract

Classical federated learning (FL) assumes that the clients have a limited amount of noisy data with which they voluntarily participate and contribute towards learning a global, more accurate model in a principled manner. The learning happens in a distributed fashion without sharing the data with the center. However, these methods do not consider the *incentive* of an agent for participating and contributing to the process, given that data collection and running a distributed algorithm locally is costly for the clients. This question of *rationality of contribution* has been asked recently in the literature and some results exist that consider this problem (Murhekar *et al.*, 2023; Karimireddy *et al.*, 2022). This paper addresses the question of simultaneous *parameter learning* and *incentivizing contribution*, which distinguishes it from the extant literature. Our first mechanism incentivizes each client to contribute to the FL process at a Nash equilibrium and simultaneously learns the model parameters. However, this equilibrium outcome can be away from the *optimal*, where clients contribute with their *full* data and the algorithm learns the *optimal* parameters. We propose a second mechanism with monetary transfers that is *budget balanced* and enables the full data contribution along with optimal parameter learning. Experiments show that these algorithms converge quite fast in practice and yield positive utility for everyone.

1 Introduction

A high quality machine learning model is built when the model is trained on a large amount of data. However, in various practical situations, e.g., for languages, images, disease modeling, such data is split across multiple entities. Moreover, in many applications, data lies in edge devices and a model is learnt when these edge devices interact with a parameter server. Federated Learning (FL) Konečný (2016); McMahan *et al.* (2017); Kairouz *et al.* (2021); Zhang *et al.* (2021) is a recently developed distributed learning paradigm where the edge devices are users' personal devices (like mobile phones and laptops) and FL aims to leverage on-device intelligence. In FL, there is a center (also known as the parameter server) and several agents (edge devices). The edge devices can only communicate through the center. However, a major challenge of federated learning is *data-scarcity*, i.e., owing to its limited storage capacity, the edge devices possesses only a few amounts of data, which may not be sufficient for the learning task. Hence, the end user participates in a distributed learning process, where she exploits the data of similar users present in the system.

In the classical federated learning problem, it is typically assumed that the agents participate with all of their data-points *voluntarily*. However, in the presence of rational users, such an assumption may not always hold since sampling data is costly for agents. It may happen that some agents try to contribute very few data-points and try to *exploit* the system by learning based on others' data-points. This phenomenon is called *free-riding* (Karimireddy *et al.*, 2022) and disincentivizes honest participants to contribute their data to the FL process. Hence, naturally, a few recent works have concentrated on designing incentive mechanisms in federated learning Murhekar *et al.* (2023); Donahue and Kleinberg (2021); Blum *et al.* (2021).

One of the most successful use-cases of federated learning is hospital management system. For instance, in a given geographical region every hospital can sample the medical records from its patients for disease modeling. To learn the model reliably, each hospital needs to collect a large amount of data which is generally very expensive. Instead, they might participate in a *federated learning* (FL) process where different hospitals can train the model locally on their individual datasets but update a consolidated global model parameter. Given that different hospitals have different capabilities in collecting data, it may be best for certain hospitals not to sample any data and therefore not incur any cost, yet get the learning parameters from the FL process if there is a sufficient population in the FL who are sharing their learned parameters. This occurrence of *free-riding* is not ideal, and an important question is to ask:

"Can a mechanism be designed that incentivizes each user to participate and contribute their maximum data in FL and simultaneously learn the optimal model parameters?"

In this paper, we address this question in two stages. First, we consider a utility model of the agents where the accuracy of learning a parameter and the contribution levels of all the agents give every agent some benefit, while their individual contribution levels lead to their personal costs. We propose an algorithm namely Updated Parameter Best Response Dynamics (`UPBReD`) in the federated learning setting that achieves simultaneous contribution and learning of the model parameters by the agents. However, this method suffers from a sub-optimal contribution and therefore the learning is also sub-optimal. In the second stage, we improve this dynamics by allowing monetary transfer where *contributors* and *consumers* are treated differently using the monetary transfer and incentivizes all agents to contribute their maximum amount of data to the learning process. This mechanism, namely Two Phase Updated Parameter Best Response Dynamics (`2P-UPBReD`), learns the optimal parameters for all agents and asks the *consumers to pay* and *pays the contributors* without keeping any *surplus*. `2P-UPBReD` is distinguished in the fact that it simultaneously learns the optimal model parameters, incentivizes full contribution from the agents, yet is quite simple to use in practice.

1.1 Related work

Federated Learning Konečnỳ (2016) has gained significant attention in the last decade or so. The success story of FL is primarily attributed to the celebrated FedAvg algorithm by McMahan *et al.* (2017), where one of the major challenges of FL, namely communication cost is reduced by *local steps*. In subsequent works, several other challenges of FL, such as data heterogeneity Karimireddy *et al.* (2020b); Ghosh *et al.* (2020a), byzantine robustness Yin *et al.* (2018); Karimireddy *et al.* (2020a); Ghosh *et al.* (2021), communication overhead Stich *et al.* (2018); Karimireddy *et al.* (2019); Ghosh *et al.* (2020b) and privacy Wei *et al.* (2020); Truex *et al.* (2020) were addressed.

Later works focused on incentivizing client participation with advanced aggregation methods. Cho *et al.* (2022) proposed dynamically weighting client updates to ensure the global model outperforms local models. Tastan *et al.* (2024) used the Shapley value to assess contributions, while Karimireddy *et al.* (2022) applied contract theory to maximize fairness in data contribution. Gao *et al.* (2021) addressed fairness by eliminating malicious clients and rewarding those who enhance performance of the model.

Monetary incentives were explored by Yu *et al.* (2020), who dynamically allocated budgets to maximize utility and minimize inequality. Blockchain-based approaches like Pandey *et al.* (2022) incentivized high-quality data contributions under budget constraints, and Yang *et al.* (2023) rewarded clients to optimize final model utility. Georgoulaki and Kollias (2023) study arbitrary utility-sharing in federated learning games, showing it achieves a price of anarchy of 2 and price of stability 1 with budget-balanced payments. Tang and Yu (2023) and Tang and Yu (2024) use Reinforcement Learning techniques in auction based FL systems with multiple centers the latter optimizes the centers' budget to maximize total utility and minimize waiting time while the former optimizes the agent bids to maximize the accumulated profit while still producing accurate trained models.

Donahue and Kleinberg (2021) modeled FL as a coalitional game, allowing agents to form coalitions for joint model training. Ray Chaudhury *et al.* (2022) uses a novel technique CoreFed to train a predictor(model) that is core stable i.e. no agents are incentivized to leave the current FL system and form a different system. Chaudhury *et al.* (2024) improves/generalizes the previous results by using ordinal utilities instead of cardinal ones. However, they did not address non-cooperative scenarios where agents strategically select data contributions.

Mao *et al.* (2024) model FL as a repeated game where agents can introduce perturbations in their data. The subgame perfect equilibrium of this game is not socially efficient and they propose a budget balanced mechanism that is socially efficient and individually rational. Blum *et al.* (2021) introduced incentive-aware learning, establishing Nash and envy-free equilibria. Murhekar *et al.* (2023) extended this with budget-balanced payments to achieve Nash equilibrium through *best response dynamics*. The budget balanced mechanism they design achieves the maximum social welfare under agent budget constraints at a Nash equilibrium. However, they did not account for model performance simultaneously with the data contribution – this is the gap that our work addresses by considering strategic agent behavior in model training.

1.2 Our contributions

We consider a strategic federated learning problem where the clients (agents) contribute by sampling data from their data distribution. The quantum of sampling is chosen such that it maximizes their individual *utilities* that consist of two opposing forces: (i) individual benefit and (ii) costs (of data sampling). Moreover, similar to the classical federated learning setting, the center aims to learn an overall model to maximize *total* accuracy (to be defined shortly) simultaneously. Our contributions can be summarized as follows.

- We propose a mechanism called Updated Parameter Best Response Dynamics, `UPBReD` (Algorithm 1) that allows simultaneous learning and contribution by the agents in FL. This mechanism does *not* use any monetary transfers.
- We show that `UPBReD` converges to a *pure strategy Nash equilibrium* (Theorem 1).
- However, the Nash equilibrium can be different from the *socially optimal* outcome where agents contribute with their full dataset $s^{\max}$ and the center learns the optimal model parameters $w^{\texttt{OPT}}$ (Example 1).
- We then propose an updated and cleaner two-phase mechanism namely Two Phase Updated Parameter Best Response Dynamics, `2P-UPBReD` (Algorithm 2) that allows monetary transfers *only* among the agents, i.e., budget balanced. The *data contributors* (those

who contribute above average quantity of data) get paid and the *data consumers* (those who contribute below average quantity of data) make the payments. However, both types of agents learn the optimal model parameters.

- We show that the Nash equilibrium for 2P-UPBReD now shifts to where all agents make their maximum contribution of data $s^{\max}$ and learn the optimal model parameters $w^{\texttt{OPT}}$ (Theorem 2).
- Experiments show that UPBReD is suboptimal on average and 2P-UPBReD converges faster than that for realistic datasets (Figure 2). Even though data consumers are required to pay in 2P-UPBReD, they still obtain positive utilities (Figure 3).

2 Preliminaries

Consider a federated learning setup where a set of data contributors, given by $N = \{1, 2, \dots, n\}$, is interacting with a center. Each data contributor (agent) i has access to a private labeled dataset D_i, with the size of the dataset given by $s_i^{\max} = |D_i|$. The agents are interested in learning a parameter vector $w \in \mathbb{R}^m$ from these data so that it helps them predict some unlabeled data accurately (e.g., to perform a classification task). However, each individual agent has a limited amount of data, and learning w only from that data may not be accurate enough. So, they learn this parameter via a *federated learner* such that the model is trained on the consolidated data of all the n users. Assume that the datasets are drawn from the same distribution, e.g., all the agents sample human disease data from a certain geographical location. However, sampling such data is costly and agent i incurs a cost $c_i(s_i)$ when she trains the model locally on her dataset of size $s_i \in S_i := [0, s_i^{\max}]$. Under this setup, each agent i gets a utility based on how much data s_i she decides to sample and train on. Hence, the utility is given by

$$u_i(w, s_i, s_{-i}) = a_i(w, s_i, s_{-i}) - z_i(s_i, s_{-i}), \tag{1}$$

where s_{-i} is the data chosen by the agents other than i in this federated learning process. The function $a_i(w, s_i, s_{-i})$ is the *accuracy function*, which denotes the *benefit* to agent i if the parameter learned by the center is w and the agents contribute by running the federated learning algorithm on their dataset sizes given by the vector $s = (s_i, s_{-i})$. The function $z_i(s_i, s_{-i})$ is the *effective cost* to agent i which may depend on the data contributions of all agents. In the first part of this paper, we consider the case where the cost is only personal to agent i, i.e., z_i is only a function of agent i's chosen data size s_i. In the second part, we assume that this cost may be additionally *taxed* or *subsidized* by the center based on their contributed data-sizes.

Since the agents are strategic, every agent i's aim is to maximize her utility by appropriately choosing her strategy s_i given the strategies s_{-i} of the other players and the parameter w chosen by the center. The center, on the other hand, is interested in learning the parameter w that maximizes the sum of the accuracies as follows, when all agents contribute their maximum data-sizes, i.e., $s_i^{\max}, i \in N$.

$$w^{\texttt{OPT}} \in \operatorname*{argmax}_{w} \sum_{i \in N} a_i(w, s_i^{\max}, s_{-i}^{\max}). \tag{2}$$

Note that the goal of the center does not consider the effective costs of the agents since those are incurred by the agents. We will refer to the term $\sum_{i \in N} a_i(w, s_i, s_{-i})$ as the *social welfare* in this context. We are interested in the question that whether we can design a federated learning algorithm that can make $(s_i^{\max}, s_{-i}^{\max})$ a Nash equilibrium of the underlying game and the center can learn $w^{\texttt{OPT}}$. In this context, the Nash equilibrium is defined as follows.

Definition 1 (Nash equilibrium). A *pure strategy Nash equilibrium* (PSNE) for a given parameter w is a strategy profile (s_i^*, s_{-i}^*) of the agents such that

$$u_i(w, s_i^*, s_{-i}^*) \geqslant u_i(w, s_i, s_{-i}^*), \ \forall s_i \in S_i, \forall i \in N.$$

This definition is a modification of the standard definition of Nash equilibrium since the equilibrium profile (s_i^*, s_{-i}^*) depends on the parameter w (we do not write it explicitly for notational cleanliness). Existence of such a PSNE in this context is obvious due to Nash's theorem (Nash Jr, 1950). To see this, consider another game Γ' where the pure strategies are to pick 0 or $s_i^{\max}$ for every $i \in N$. Every mixed strategy of Γ' is a pure strategy of the original data contribution game we consider above. Since by Nash's theorem, a mixed strategy Nash equilibrium exists for any finite game, PSNE exists in our game. We will refer to the term PSNE as Nash equilibrium (NE) in the rest of the paper.

We also aim for the following property of a mechanism that involve monetary transfers.

Definition 2 (Budget balance). A mechanism that uses monetary transfers $p_i(s_i, s_{-i})$ for every $s_i \in S_i, i \in N$ is called *budget balanced* (BB) if $\sum_{i \in N} p_i(s_i, s_{-i}) = 0$.

This property ensures that the net monetary in or out-flow is zero and the mechanism only allows monetary redistribution among the agents. In the following section, we consider the general case where the effective cost is arbitrary.

3 Learning with arbitrary effective costs

In this section, we consider the effective cost of agent i, i.e., $z_i(s_i, s_{-i})$, to be an arbitrary function of $s = (s_i, s_{-i})$. We know that an alternative interpretation of NE (Definition 1) is a strategy profile (s_i^*, s_{-i}^*) where every agent's *best response* to the strategies of the other players is its own strategy in that profile (see Maschler *et al.* (2020, e.g.)), i.e.,

$$s_i^* \in \underset{s_i \in S_i}{\operatorname{argmax}} \ u_i(w, s_i, s_{-i}^*), \ \forall i \in N.$$

Hence, an algorithm that simultaneously updates all agents' strategies with the best responses to the current strategies of the other players is called a *best response dynamics* of a strategic form game (see Fudenberg (1991, e.g.) for a detailed description). In our problem, this approach cannot be directly employed since the center also needs to learn and update the model parameters w as the agents choose their data contributions. We, therefore, propose a mechanism for federated learning that simultaneously updates both the agents' strategies and the center's choice of w.

In this mechanism, each agent i starts with some initial choices of s_i^0 and shares that with the center.[1] The center also starts with a w^0 and broadcasts that and the entire initial data-contribution vector s^0 with all the agents. In every subsequent iteration t, each agent i locally computes two quantities: (i) her updated contribution s_i^{t+1} by taking one gradient ascent step w.r.t. her own contribution s_i, and (ii) the local gradient of agent i's accuracy component of the social welfare w.r.t. w. Both are evaluated at the current values of w^t and s^t and sent back to the center. The center calculates an updated w^{t+1} that averages (in spirit of the FedAvg algorithm (McMahan *et al.*, 2017)) all the local gradients sent by the agents. The center then shares w^{t+1} and s^{t+1} with all the agents. Formally, the updates are given as follows.

$$s^{t+1} = s^t + \gamma g(w^t, s^t, \mu^t), \tag{3}$$
$$w^{t+1} = w^t + \eta \tilde{g}(w^t, s^t), \tag{4}$$

[1]Note that, only the fraction s_i is shared with the center and not the data, which is consistent with the principle of federated learning.

Algorithm 1 Updated Parameter Best Response Dynamics (`UPBReD`)

Require: Step size γ, η, initialization w^0, $s_i^0, i \in N$, number of iterations T.

for $t = 0, 1, \dots, T-1$ **do**
    Center: broadcasts w^t, s^t
    for agent $i \in N$ in parallel **do**
        $s^{t+1} = s^t + \gamma g(w^t, s^t, \mu^t)$
        Compute local gradient: $d_i^{t+1} = \nabla_w a_i(w^t, s_i^{t+1}, s_{-i}^t)$, send s_i^{t+1}, d_i^{t+1} to the center
    end for
    Center:
    update: $w^{t+1} = w^t + \frac{\eta}{n} \sum_{i \in N} d_i^{t+1} = w^t + \eta \tilde{g}(w^t, s^t)$,
end for
return w^T

where the functions g and $\tilde{g}$ are defined as:

$$[g(w^t, s^t, \mu^t)]_i = \frac{\partial}{\partial s_i} u_i(w^t, s^t) + \mu_i^t, \text{ with} \tag{5}$$

$$\mu_i^t = \begin{cases} -\frac{\partial}{\partial s_i} u_i(w^t, s^t), & \text{when either } s^t = 0, \frac{\partial}{\partial s_i} u_i < 0 \\ & \text{or } s^t = s_i^{\max}, \frac{\partial}{\partial s_i} u_i > 0 \\ 0. & \text{otherwise} \end{cases}$$

$$\tilde{g}(w^t, s^t) = \frac{1}{n} \sum_{i \in N} \nabla_w a_i(w^t, s^t). \tag{6}$$

The mechanism is detailed out in Algorithm 1.

Our main result of this section is that under certain concavity and bounded derivative conditions, Algorithm 1 always converges to a Nash equilibrium. We need a few matrices, defined as follows for every w and s, to present the result.

$$\begin{aligned} G(w,s)_{ij} &= \frac{\partial^2}{\partial s_j \partial s_i} u_i(w,s), \quad i, j \in N; \\ \tilde{G}(w,s)_{k\ell} &= \frac{1}{n} \sum_{i \in N} \frac{\partial^2}{\partial w_\ell \partial w_k} a_i(w,s), \quad k, \ell \in \{1, \dots, m\}; \\ H(w,s)_{ik} &= \frac{\partial^2}{\partial w_k \partial s_i} u_i(w,s), \quad i \in N, k \in \{1, \dots, m\}; \\ \tilde{H}(w,s)_{kj} &= \sum_{i \in N} \frac{\partial^2}{\partial s_j \partial w_k} a_i(w,s), \quad j \in N, k \in \{1, \dots, m\}. \end{aligned} \tag{7}$$

In the following, we state the assumptions.

Assumption 1. Consider the utility functions given by Equation (1) where functions $a_i, z_i, i \in N$ are such that the following properties hold for every $w \in \mathbb{R}^m$ and $s \in \prod_{i \in N} S_i$ (the matrices below are as defined in Equation (7)).

1. The matrices $G(w,s) + \lambda \mathbb{I}$ and $\tilde{G}(w,s) + \tilde{\lambda} \mathbb{I}$ are negative semi-definite.
2. We assume the following bounds:
    (a) $\forall i, j \in N, |G(w,s)_{ij}| \leqslant L$,
    (b) $\forall k, \ell \in \{1, \dots, m\}, |\tilde{G}(w,s)_{k\ell}| \leqslant \tilde{L}$,
    (c) $\|H(w,s)\|_{op} \leqslant P$, where $\|A\|_{op} := \inf\{c \geqslant 0 : \|Av\| \leqslant c\|v\|, \forall v\}$,
    (d) $\|\tilde{H}(w,s)\|_{op} \leqslant \tilde{P}$.

Note that the definition of G and $\tilde{G}$ do not imply the strong concavity of u_i or a_i. Moreover, the above assumptions are standard and appeared in the literature before (see Murhekar *et al.* (2023)). We define the following expressions for a cleaner presentation.

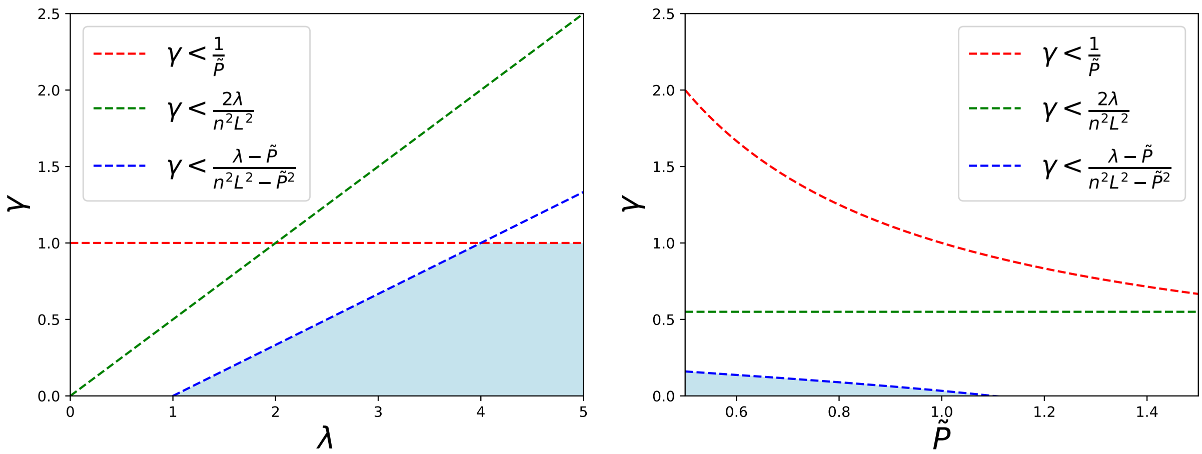
$$\begin{aligned}
W_1 &= \sqrt{1+\gamma^2 n^2 L^2 - 2\gamma\lambda} + \sqrt{\tilde{P}^2\gamma^2}, \\
W_2 &= \sqrt{1+\eta^2 m^2 \tilde{L}^2 - 2\eta\tilde{\lambda}} + \sqrt{P^2\eta^2}, \\
E &= \|g(w^0, s^0, \mu^0)\|_2 + \|\tilde{g}(w^0, s^0)\|_2, \\
&\text{where } w^0, s^0 \text{ are arbitrary, and} \\
T_0(w^0, s^0) &= \left(\ln \frac{E}{\epsilon}\right) / \left(\ln \frac{1}{W}\right), \text{ where } W = \max\{W_1, W_2\}.
\end{aligned} \tag{8}$$

We are now ready to present the main result of this section.

Theorem 1 (`UPBReD` convergence to NE)**.** *Under Assumption 1,* `UPBReD` *(Algorithm 1) converges to a Nash equilibrium. Formally, for a given $\epsilon > 0$, for every initial value (w^0, s^0) of Algorithm 1, the gradients $\|g(w^T, s^T, \mu^T)\| < \epsilon$ and $\|\tilde{g}(w^T, s^T)\| < \epsilon$, for all $T \geqslant T_0(w^0, s^0)$, when the step sizes are chosen as follows*

$$\gamma < \min\left\{1, \tfrac{1}{\tilde{P}}, \tfrac{2\lambda}{n^2L^2}, \tfrac{\lambda-\tilde{P}}{n^2L^2-\tilde{P}^2}\right\}, \text{ given } \lambda > \tilde{P},$$
$$\eta < \min\left\{1, \tfrac{1}{P}, \tfrac{2\tilde{\lambda}}{m^2\tilde{L}^2}, \tfrac{\tilde{\lambda}-P}{m^2\tilde{L}^2-P^2}\right\}, \text{ given } \tilde{\lambda} > P.$$

The theorem provides a guarantee of convergence of Algorithm 1 for any arbitrary initial condition (w^0, s^0). It needs to run for a minimum number of iterations $T_0(w^0, s^0)$, with appropriate parameters of γ and η that govern the gradient ascent rates of the agents' and center's objectives respectively. Note that for a better convergence, i.e., a smaller ϵ, the algorithm needs to run longer. The parameter W determines the contraction factor of the recurrence of $\|g(w^t, s^t)\|_2 + \|\tilde{g}(w^t, s^t)\|_2$ and is chosen to be smaller than unity by choosing γ and η appropriately. Figure 1 shows feasible regions of γ for certain $n, L, \lambda, \tilde{P}$.



(a) $\tilde{P} = 1$

(b) $\lambda = 1.1$

Figure 1: *Shaded regions show the feasible choices of γ for $n = 2, L = 1$ and $\tilde{P}$ and λ as shown. The dashed lines show the boundary of the regions in the legends. A similar set of choices is true for η.*

Remark 1. Note that the choice of step sizes γ and ν scale inversely with m and n respectively in the above theorem. Moreover, we impose certain restrictions on λ and $\tilde{\lambda}$, which may be restrictive. In Section 4, we remove such issue and propose a learning algorithm with choices of γ and ν that do not scale with m or n.

Remark 2. We emphasize that in experiments, sufficiently small and constant values of γ and ν work. Hence, the implicit choice obtained in the above theorem is an artifact of our theoretical analysis, and is not a reflection of practical scenarios.

Proof. (of Theorem 1) Consider the first-order Taylor expansions of g and $\tilde{g}$ given as follows.

$$\begin{aligned} g(w^{t+1}, s^{t+1}, \mu^{t+1}) &= g(w^t, s^t, \mu^t) + G(w^t, s', \mu^t) \cdot (s^{t+1} - s^t) \\ &\quad + H(w', s^t, \mu^t)(w^{t+1} - w^t), \\ \tilde{g}(w^{t+1}, s^{t+1}) &= \tilde{g}(w^t, s^t) + \tilde{G}(w', s^t) \cdot (w^{t+1} - w^t) \\ &\quad + \tilde{H}(w^t, s')(s^{t+1} - s^t), \end{aligned}$$

where $s' = \theta_s s^t + (1 - \theta_s)s^{t+1}, w' = \theta_w w^t + (1 - \theta_w)w^{t+1}$, for some $\theta_s, \theta_w \in [0, 1]$.

Using the update rules of Algorithm 1 given by Equations (3) and (4), we get

$$\begin{aligned} g(w^{t+1}, s^{t+1}, \mu^{t+1}) &= g(w^t, s^t, \mu^t) + G(w^t, s', \mu^t) \cdot \gamma g(w^t, s^t, \mu^t) \\ &\quad + H(w', s^t, \mu^t)\eta\tilde{g}(w^t, s^t) \\ \tilde{g}(w^{t+1}, s^{t+1}) &= \tilde{g}(w^t, s^t) + \tilde{G}(w', s^t) \cdot \eta\tilde{g}(w^t, s^t) \\ &\quad + \tilde{H}(w^t, s')\gamma g(w^t, s^t, \mu^t) \end{aligned}$$

Using triangle inequality on each of these identities, we get

$$\begin{aligned} \|g(w^{t+1}, s^{t+1}, \mu^{t+1})\|_2 &\leqslant \|(I_{n\times n} + \gamma G(w^t, s', \mu^t))g(w^t, s^t, \mu^t)\|_2 \\ &\quad + \eta\|H(w', s^t, \mu^t)\tilde{g}(w^t, s^t)\|_2 \\ \|\tilde{g}(w^{t+1}, s^{t+1})\|_2 &\leqslant \|(I_{m\times m} + \eta\tilde{G}(w', s^t))\tilde{g}(w^t, s^t)\|_2 \\ &\quad + \gamma\|\tilde{H}(w^t, s')g(w^t, s^t, \mu^t)\|_2 \end{aligned} \tag{9}$$

From condition 1 of Assumption 1, we get that $v^\top(G + \lambda I_{n\times n})v \leqslant 0, \forall v \in \mathbb{R}^n$ and $v'^\top(\tilde{G} + \lambda I_{m\times m})v' \leqslant 0, \forall v' \in \mathbb{R}^m$. In particular, for $v = g(w^t, s^t, \mu^t)$ and $v' = \tilde{g}(w^t, s^t)$, we get

$$\begin{aligned} g(w^t, s^t, \mu^t)^\top G(w^t, s', \mu^t)g(w^t, s^t, \mu^t) &\leqslant -\lambda\|g(w^t, s^t, \mu^t)\|_2^2 \\ \tilde{g}(w^t, s^t)^\top \tilde{G}(w', s^t)\tilde{g}(w^t, s^t) &\leqslant -\tilde{\lambda}\|\tilde{g}(w^t, s^t)\|_2^2 \end{aligned} \tag{10}$$

Consider the square of the first term of the RHS of the inequality of g in Equation (9)

$$\begin{aligned} &\|(I_{n\times n} + \gamma.G(w^t, s', \mu^t))g(w^t, s^t, \mu^t)\|^2 \\ &= \|g(w^t, s^t, \mu^t)\|_2^2 + \gamma^2.\|G(w^t, s^t)g(w^t, s^t, \mu^t)\|_2^2 \\ &\quad + 2\gamma g(w^t, s^t, \mu^t)^\top G(w^t, s', \mu^t)g(w^t, s^t, \mu^t) \\ &\leqslant \|g(w^t, s^t, \mu^t)\|_2^2 + \gamma^2.n^2L^2\|g(w^t, s^t, \mu^t)\|_2^2 - 2\gamma\lambda\|g(w^t, s^t, \mu^t)\|_2^2 \\ &= (1 + \gamma^2.n^2L^2 - 2\gamma\lambda)\|g(w^t, s^t, \mu^t)\|_2^2 \end{aligned} \tag{11}$$

where the equality comes by expanding the squared norm and the inequality comes from the facts that

(i) $\|G(w^t, s^t)g(w^t, s^t, \mu^t)\|_2^2 \leqslant \|G(w^t, s^t)\|_F^2\|g(w^t, s^t, \mu^t)\|_2^2$, where $\|A\|_F := \sqrt{\sum_i\sum_j |A_{ij}|^2}$ is the Frobenius norm of a matrix A and (ii) using Equation (10). By condition 2 of Assumption 1, $\|G(w^t, s^t)\|_F^2 \leqslant n^2L^2$. Hence, we get

$$\begin{aligned} &\|(I_{n\times n} + \gamma.G(w^t, s^t))g(w^t, s^t, \mu^t)\| \\ &\leqslant \sqrt{1 + \gamma^2.n^2L^2 - 2\gamma\lambda} \cdot \|g(w^t, s^t, \mu^t)\|_2 \end{aligned}$$

given the term inside the square root is positive.

Consider the second term of the RHS of the inequality of g in Equation (9), where

$$\begin{aligned}\|H(w', s^t, \mu^t)\tilde{g}(w^t, s^t)\|_2^2 &\leqslant \|H(w', s^t, \mu^t)\|_{op}^2 \|\tilde{g}(w^t, s^t)\|_2^2 \\ &\leqslant P^2 \|\tilde{g}(w^t, s^t)\|_2^2\end{aligned}$$

by condition 2 of Assumption 1.

Defining $\alpha := 1 + \gamma^2.n^2L^2 - 2\gamma\lambda, \beta := P^2\eta^2, \tilde{\alpha} := 1 + \eta^2.m^2\tilde{L}^2 - 2\eta\tilde{\lambda}, \tilde{\beta} := \tilde{P}^2\gamma^2$, and carrying out a similar analysis for $\tilde{g}$ in Equation (9), we get

$$\begin{aligned}\|g(w^{t+1}, s^{t+1}, \mu^{t+1})\|_2 &\leqslant \sqrt{\alpha}\|g(w^t, s^t, \mu^t)\|_2 + \sqrt{\beta}\|\tilde{g}(w^t, s^t)\|_2 \\ \|\tilde{g}(w^{t+1}, s^{t+1})\|_2 &\leqslant \sqrt{\tilde{\alpha}}\|\tilde{g}(w^t, s^t)\|_2 + \sqrt{\tilde{\beta}}\|g(w^t, s^t, \mu^t)\|_2\end{aligned} \tag{12}$$

Adding the inequalities of Equation (12)

$$\begin{aligned}&\|g(w^{t+1}, s^{t+1}, \mu^{t+1})\|_2 + \|\tilde{g}(w^{t+1}, s^{t+1})\|_2 \\ &\leqslant \max\{\sqrt{\alpha} + \sqrt{\tilde{\beta}}, \sqrt{\tilde{\alpha}} + \sqrt{\beta}\} \left(\|g(w^t, s^t, \mu^t)\|_2 + \|\tilde{g}(w^t, s^t)\|_2\right)\end{aligned} \tag{13}$$

To ensure that the above inequality is a contraction, we need to ensure $\sqrt{\alpha} + \sqrt{\tilde{\beta}}, \sqrt{\tilde{\alpha}} + \sqrt{\beta} \in (0, 1)$. These imply (i) $0 < \alpha < 1, 0 < \tilde{\alpha} < 1$, (ii) $\sqrt{\tilde{\beta}} < 1, \sqrt{\beta} < 1$, and (iii) $\sqrt{\alpha} + \sqrt{\tilde{\beta}} < 1, \sqrt{\tilde{\alpha}} + \sqrt{\beta} < 1$. We can solve for γ and η from these inequalities and obtain the sufficient conditions when $\lambda > \tilde{P}$ and $\tilde{\lambda} > P$:

$$\begin{aligned}&\gamma < \tfrac{1}{\tilde{P}}, \gamma < \tfrac{2\lambda}{n^2L^2}, \text{ and } \gamma < \tfrac{\lambda - \tilde{P}}{n^2L^2 - \tilde{P}^2}, \\ &\eta < \tfrac{1}{P}, \eta < \tfrac{2\tilde{\lambda}}{m^2\tilde{L}^2}, \text{ and } \eta < \tfrac{\tilde{\lambda} - P}{m^2\tilde{L}^2 - P^2}.\end{aligned}$$

Notice that from conditions 1 and 2 of Assumption 1, we have that $\lambda < L$ and $\tilde{\lambda} < \tilde{L}$. Thus $n^2L^2 - \tilde{P}^2 > 0$ and $m^2\tilde{L}^2 - P^2 > 0$. For the γ, η chosen as above, we find $W_1 = \sqrt{\alpha} + \sqrt{\tilde{\beta}} < 1$ and $W_2 = \sqrt{\tilde{\alpha}} + \sqrt{\beta} < 1$, and hence $W = \max\{W_1, W_2\} < 1$. Therefore,

$$\begin{aligned}&\|g(w^{t+1}, s^{t+1}, \mu^{t+1})\|_2 + \|\tilde{g}(w^{t+1}, s^{t+1})\|_2 \\ &\leqslant W\left(\|g(w^t, s^t, \mu^t)\|_2 + \|\tilde{g}(w^t, s^t)\|_2\right).\end{aligned}$$

Recursively iterating over this inequality, we get

$$\begin{aligned}&\|g(w^T, s^T, \mu^T)\|_2 + \|\tilde{g}(w^T, s^T)\|_2 \\ &\leqslant W^T\left(\|g(w^0, s^0, \mu^0)\|_2 + \|\tilde{g}(w^0, s^0)\|_2\right).\end{aligned}$$

Defining $E = \|g(w^0, s^0, \mu^0)\|_2 + \|\tilde{g}(w^0, s^0)\|_2$ and

$$T_0(w^0, s^0) = \left(\ln \frac{E}{\epsilon}\right) / \left(\ln \frac{1}{W}\right),$$

we get that for all $T \geqslant T_0(w^0, s^0)$,

$$\begin{aligned}&\|g(w^T, s^T, \mu^T)\|_2 + \|\tilde{g}(w^T, s^T)\|_2 < \epsilon \\ \implies &\|g(w^T, s^T, \mu^T)\|_2 < \epsilon, \text{ and } \|\tilde{g}(w^T, s^T)\|_2 < \epsilon.\end{aligned}$$

This completes the proof. $\square$

3.1 Application: effective costs are only personal

One special but practical case of the above setup is when the effective cost is only borne by the agent. Mathematically, it is represented as $z_i(s_i, s_{-i}) = c_i(s_i), \forall s_i \in S_i, s_{-i} \in S_{-i}, \forall i \in N$.

We assume convex cost functions $c_i, \forall i \in N$, which is a commonly used assumption Li and Raghunathan (2014) in the literature. This setup is consistent with the assumptions of Assumption 1, and according to Theorem 1, Algorithm 1 converges to a Nash equilibrium, which we denote as (w^*, s^*). However, it is possible that none of the following things happen: (i) $s^* \neq s^{\max}$, i.e., the agents do not contribute their entire data for the federated learning process, leading to a suboptimal learning, or (ii) $w^* \neq w^{\texttt{OPT}}$ (see Equation (2) for the definition), which is neither the objective of the center nor the agents. The following example shows such an instance.

Example 1 (NE different from socially optimal)**.** Consider the federated learning setup with two agents that have linear costs and are trying to learn a model with two parameters. The accuracy function is identical and is given by $a_i(w, s) = 1 - \frac{(1-w_1)^2+(2-w_2)^2}{s_1+s_2}, i = 1, 2$. The cost functions for agents 1 and 2 are given by $c_1(s_1) = 0.04 \cdot s_1$ and $c_2(s_2) = 0.02 \cdot s_2$ respectively. The agents' strategy sets are $S_i = [0, 5], i = 1, 2$. Notice that, at $s = s^{\max} = (5, 5)$, the value of $w^{\texttt{OPT}}$ (see Equation (2)) is $(1, 2)$. It yields an optimal social welfare of 2. However, this is not an NE, since the derivatives $\partial u_i / \partial s_i|_{s_i^{\max}} < 0$ for both $i = 1, 2$. Hence, the best response of each agent is to reduce s_i. However, from this point, with the choice of $\gamma = 0.25$ and $\eta = 0.25$, Algorithm 1 converges to an NE profile of $w^* = (0.5, 1.5), s^* = (0, 5)$ that yields a social welfare of 1.8. □

So, our objective in this paper is to allow monetary transfer among the agents so that we get the best of both worlds: (a) We achieve convergence to a Nash equilibrium of $(w^{\texttt{OPT}}, s^{\max})$, where all agents to contribute their entire data and center learns the optimal parameter, and (b) the transfers are *budget balanced*, i.e., the center does not accumulate any money – it is used just to realign the agents' utilities to reach the desired Nash equilibrium. We discuss this mechanism in the following section.

4 Welfare maximization with monetary transfers

In this section, we consider the scenario where the center can make a payment $p_i(s_i, s_{-i})$ *to* agent i to alter her utility function. Therefore, the effective cost of agent i becomes

$$z_i(s_i, s_{-i}) = c_i(s_i) - p_i(s_i, s_{-i}). \tag{14}$$

Note that, the payment can be either positive or negative, which determines whether the agent is *subsidized* or *taxed* respectively. It is reasonable to expect that the *data contributors* of the federated learning process are subsidized for their data contribution while the *data consumers* are charged payment for obtaining the learned parameter. We assume that the accuracy function a_i is monotone non-decreasing in s_i. This assumption captures the fact that the benefit to an agent increases when she contributes more data to the learning process. We choose the following payment function:

$$p_i(s) = \beta \left(s_i - \frac{1}{n-1} \sum_{j \neq i} s_j \right), \tag{15}$$

where β is a parameter of choice. Note that this payment mechanism is budget balanced by design, since $\sum_{i \in N} p_i(s_i, s_{-i}) = 0, \forall s_i \in S_i, s_{-i} \in S_{-i}$. With payment, the utility of agent i

becomes:

$$u_i(w, s) = a_i(w, s) - c_i(s_i) + \beta \left(s_i - \frac{1}{n-1} \sum_{j \neq i} s_j \right). \tag{16}$$

Since we chose the payment to be *linear* in s, and $c_i(s_i)$ is convex in s_i, the effective payment z_i is convex in s. However, in this section, we are also interested in the *quality* of the NE in terms of social welfare and want to reach the desired NE where $s_i^* = s_i^{\max}, \forall i \in N$ and $w^* = w^{\mathtt{OPT}}$.

4.1 Convergence to the NE $(w^{\mathrm{OPT}}, s^{\max})$

In this setup, we first prove that the utility of every agent i can be made strictly increasing in her own contribution s_i in the following manner.

Lemma 1 (Increasing utility). *If the accuracy function a_i is monotone non-decreasing in s_i for all $i \in N$, there exists a payment function given by Equation (15) that ensures that the utility of every agent is increasing in her own data contribution, i.e., $\frac{\partial}{\partial s_i} u_i(w, s_i, s_{-i}) > 0, \forall s_i \in (0, s_i^{\max}), \forall i \in N, \forall w$.*

Proof. From Equation (16), we get the derivative of the utility function for $s_i \in (0, s_i^{\max})$ (note that this holds only for the interior of S_i, the derivative at the boundaries are zero by definition of Equation (5)) to be $[g(w, s, \mu)]_i = \frac{\partial}{\partial s_i} u_i(w, s) = \frac{\partial}{\partial s_i} a_i(w, s) - c_i'(s_i) + \beta$. Since the cost function is convex, its derivative is increasing in s_i and hence we have $c_i'(s_i) \leqslant c_i'(s_i^{\max})$, $\forall s_i \in S_i$ and $\forall i \in N$. Choose in Equation (15) the parameter $\beta > \max_{i \in N} c_i'(s_i^{\max})$, where $c_i'(s_i) = \frac{d}{ds_i} c_i(s_i)$. Therefore

$$\begin{aligned} [g(w, s, \mu)]_i &= \tfrac{\partial}{\partial s_i} a_i(w, s) - c_i'(s_i) + \beta \\ &\geqslant \tfrac{\partial}{\partial s_i} a_i(w, s) - c_i'(s_i^{\max}) + \beta > 0. \end{aligned}$$

The last inequality holds since we have $\frac{\partial}{\partial s_i} a_i(w, s) \geqslant 0$ and $\beta > c_i'(s_i^{\max})$, $\forall i \in N$. □

Remark 3. Throughout this paper, we use $\beta > \max_{i \in N} c_i'(s_i^{\max})$.

This lemma implies that even in Algorithm 1 if we apply the above payment, s will converge to the maximum value $s^{\max}$. However, unlike Algorithm 1, in this section we provide an algorithm which gives the step sizes of the gradient ascents in a more concrete manner and is, therefore, a superior one. In order to keep the convergence rate same as Theorem 1, we assume that the negative[2] social welfare function at $s^{\max}$, given by $f(w, s^{\max}) = -\frac{1}{n} \sum_{i \in N} a_i(w, s^{\max})$, is M-smooth and ν-strictly convex in w. These properties are formally defined below.

Definition 3 (M-smoothness). A function $f : \mathbb{R}^m \to \mathbb{R}$ is M-smooth if for all $x, x' \in \mathbb{R}^m$

$$f(x') \leqslant f(x) + \langle \nabla_x f(x), x' - x \rangle + \frac{M}{2} \|x - x'\|_2^2.$$

Definition 4 (ν-strictly convex). A function $f : \mathbb{R}^m \to \mathbb{R}$ is ν-strictly convex if for all $x, x' \in \mathbb{R}^m$

$$f(x') \geqslant f(x) + \langle \nabla_x f(x), x' - x \rangle + \frac{\nu}{2} \|x - x'\|_2^2.$$

We propose a two-phase algorithm given by Algorithm 2. The algorithm, in the first phase, incentivizes the agents to contribute $s^{\max}$, and in the second phase, converges to $w^{\mathtt{OPT}}$. Note that in the first phase of Algorithm 2, every agent runs a gradient ascent step. Thanks to the increasing utility property (Lemma 1), we have an increasing sequence of s_i^t for all $i \in N$.

[2] We negate the welfare so that we can consider functions as convex to apply the results of convex analysis easily.

Algorithm 2 Two Phase Updated Parameter Best Response Dynamics (2P-UPBReD)

Require: Step size γ, η, initialization w^0, $s_i^0, i \in N$, number of iterations T.

Center: broadcasts w^0, s^0, set $t = 0$ ▷ begin phase 1
while $s \neq s^{\max}$ **do**
  Center: broadcasts s^t
  for agent $i \in N$ in parallel **do**
    $s_i^{t+1} = s_i^t + \gamma[g(w^0, s^t, \mu^t)]_i$
    Sends s_i^{t+1} to the center
  end for
  $t = t + 1$
end while ▷ end phase 1
for $t = 0, 1, \ldots, T$ **do** ▷ begin phase 2
  Center: broadcasts w^t
  for agent $i \in N$ in parallel **do**
    Compute local gradient: $d_i^{t+1} = \nabla_w a_i(w^t, s_i^{\max}, s_{-i}^{\max})$, send d_i^{t+1} to the center
  end for
  Center:
  update: $w^{t+1} = w^t + \frac{\eta}{n} \sum_{i \in N} d_i^{t+1} = w^t + \eta \tilde{g}(w^t, s^{\max})$,
end for
return w^T ▷ end phase 2

Moreover, from the definition of $[g(w^0, s^t, \mu^t)]_i$ it is ensured that $s_i^{\max}$ is the fixed point of this gradient ascent update. Hence after a finite number of iterations, every agent reaches $s_i^{\max}$. Note that in this phase the model parameter w^0 remains unchanged.

In the second phase of the algorithm, we update the model parameter w^t. Note that since all the agents contribute the maximum amount of data they own, this phase is simply *pure* federated learning (without incentive design). As such, each agent now computes the local gradient d_i^{t+1} and sends it to the center. The center then aggregates the gradients and take a gradient ascent step with learning rate η. The center then broadcasts the updated parameter to the agents and the process continues.

We now provide the convergence guarantees of Algorithm 2. Before that, let us discuss the assumptions as discussed in Section 3.1.

Assumption 2. Consider the utility functions given by Equation (1) where functions $a_i, c_i, i \in N$ are such that the following properties hold.

1. The function $a_i(w, s_i, s_{-i})$ is monotone non-decreasing in s_i for every $w, s_{-i}, i \in N$.
2. The function $c_i(s_i)$ is convex and differentiable everywhere for $s_i \in S_i, i \in N$.
3. The negative social welfare function at $s^{\max}$, $f(w, s^{\max})$, is M-smooth and ν-strictly convex in w, with $M > \nu$.

In expected utility theory in microeconomics, cardinal utility function of *risk-averse* agents is assumed to be concave Pemberton and Rau (2011); Li and Raghunathan (2014); Murhekar *et al.* (2023) which the above assumptions imply. Also, the smoothness and strong convexity assumptions have featured in several previous papers on FL Karimireddy *et al.* (2020b); Yin *et al.* (2018). The main result of this section is as follows.

Theorem 2 (2P-UPBReD convergence to the optimal NE)**.** *Suppose Assumption 2 holds and we consider the utility function with payment scheme given in Equation (16) with $\beta > \max_{i\in N} c_i'(s_i^{\max})$. Then, for every $w \in \mathbb{R}^m$ and $s \in \prod_{i\in N} S_i$, 2P-UPBReD (Algorithm 2) converges to the Nash equilibrium $(w^{\mathtt{OPT}}, s^{\max})$ and is budget balanced. In particular, for any given $\epsilon > 0$, we have*

$$\tfrac{1}{n}\textstyle\sum_{i\in N} a_i(w^{\mathtt{OPT}}, s^{\max}) - \tfrac{1}{n}\sum_{i\in N} a_i(w^T, s^{\max}) < \epsilon$$

and $s^T = s^{\max}$ in $T = \kappa + T_0$ iterations provided we choose $\gamma = c$ (an universal constant) and $\eta = 1/M$ with

$$\kappa \geqslant \max_i \left\{ \frac{s_i^{\max} - s_i^0}{c\,\Delta_i} \right\} \tag{17}$$

$$T_0 > \left(\ln \frac{f(w^0, s^{\max}) - f(w^{\mathtt{OPT}}, s^{\max})}{\epsilon} \right) \Big/ \left(\ln \frac{1}{1 - \frac{\nu}{M}} \right) \tag{18}$$

where, $f(w,s) := -\frac{1}{n}\sum_{i\in N} a_i(w,s)$ and $\Delta_i = \beta - c_i'(s_i^{\max})$.

Proof sketch. In phase 1, we show that the utilities are strictly increasing for every agent, and therefore, the algorithm reaches the state $(w^0, s^{\max})$. Phase 2 is a gradient descent on $f(\cdot, s^{\max})$ which leads w to converge to $w^{\mathtt{OPT}}$ in T_0 iterations due to the smoothness and strict convexity of f. □

Proof. Phase 1. Note that the derivatives of the utility of agent $i \in N$ is given by

$$\begin{aligned} [g(w^0, s^t, \mu^t)]_i &= \tfrac{\partial}{\partial s_i} u_i(w^0, s^t) \\ &= \tfrac{\partial}{\partial s_i} a_i(w^0, s^t) - \tfrac{\partial}{\partial s_i} c_i(s_i) + \beta \\ &\geqslant \tfrac{\partial}{\partial s_i} a_i(w^0, s^t) - \tfrac{\partial}{\partial s_i} c_i(s_i^{\max}) + \beta. \end{aligned} \tag{19}$$

The inequality follows since c_i is convex and differentiable everywhere in S_i and its derivative is a non-decreasing function of s_i. This implies that the maximum value of $c_i(s_i)$ is reached at $s_i = s^{\max}$. For the choice of $\beta > \max_{i\in N} c_i'(s_i^{\max})$, we get $\Delta_i := \beta - c_i'(s_i^{\max}) > 0$. Since, a_i is non-decreasing in s_i, we get that the RHS of Equation (19) is positive. Hence

$$[g(w^0, s^t, \mu^t)]_i > 0,$$

implying that at every step of this phase, the utility function increases for every agent. Applying the update rule for s given by Algorithm 2 in phase 1, we get

$$s^{t+1} = s^t + \gamma g(w^0, s^t, \mu^t) \geqslant s^t + \tfrac{\partial}{\partial s_i} a_i(w^t, s^t) + \Delta_i,$$

and applying the inequality repeatedly for t iterations yields

$$s_i^t \geqslant s_i^0 + \gamma(t\Delta_i + \textstyle\sum_{k=0}^t \tfrac{\partial}{\partial s_i} a_i(w^0, s^k)).$$

Let $l_i = s_i^{\max} - s_i^0$. We have,

$$\begin{aligned} s_i^t &\geqslant s_i^{\max} - l_i + \gamma(t\Delta_i + \textstyle\sum_{k=1}^t \tfrac{\partial}{\partial s_i} a_i(s^k, w^k)) \\ &\geqslant s_i^{max} - l_i + \gamma t \Delta_i \end{aligned}$$

where non-decreasingness of a_i w.r.t. s_i is used. Substituting $t = \kappa$ where $\kappa \geqslant \frac{l_i}{\Delta_i \gamma}$, we obtain

$$\begin{aligned} s_i^\kappa &\geqslant s_i^{\max} - k_i + \gamma \Delta_i (\frac{k_i}{\Delta_i \gamma}) \\ &\geqslant s_i^{\max} - k_i + k_i \\ &\geqslant s_i^{\max} \\ s_i^\kappa &\geqslant s_i^{\max} \end{aligned} \tag{20}$$

Since, $s_i^t \leqslant s_i^{\max}$ for all t and $i \in N$, we conclude that $s_i^\kappa = s_i^{\max}$ for all $i \in N$. Hence, $s^{\max}$ is a fixed point of the update $s^{t+1} = s^t + \gamma g(w^0, s^t, \mu^t)$ at the end of phase 1.

Phase 2 For $t > \kappa$, we can focus on the second phase of the algorithm. From the fixed point property of $s^{\max}$, we can now focus entirely on the function $f(w, s^{\max}) = -\frac{1}{n} \sum_{i \in N} a_i(w, s^{\max})$. From the algorithmic description, the second phase is just a simple gradient descent, run by the center, on $f(w, s^{\max})$. This is easy to see since at every iteration the center gets d_i^{t+1} from all the agents $i \in N$, aggregates them and construct $\tilde{g}(w^t, s^{\max})$, which is the gradient of $f(w, s^{\max})$ computed at w^t. We denote $f(w, s^{\max})$ with $f(w)$ in this part for notational cleanliness.

Note that the gradient descent is run with initialization w^κ (which is same as w^0 as given by first phase of the algorithm). We exploit the strong convexity and smoothness of $f(w, s^{\max})$. Running the second phase for T_0 iterations, using Wright and Recht (2022), we obtain

$$f(w^{T_0}) - f(w^{\mathtt{OPT}}) \leq (1 - \tfrac{\nu}{M})^{T_0} [f(w^0) - f(w^{\mathtt{OPT}})].$$

Hence, for $f(w^{T_0}) - f(w^{\mathtt{OPT}}) < \epsilon$, we require

$$T_0 > \left(\ln \tfrac{f(w^0) - f(w^{\mathtt{OPT}})}{\epsilon} \right) \Big/ \left(\ln \frac{1}{\left(1 - \frac{\nu}{M}\right)} \right),$$

which proves the theorem. □

We present a few observations here.

Remark 4 (γ and rationality). Phase 1 of Algorithm 2 achieves the optimal contribution $s^{\max}$ without updating the learning parameter w. The speed of convergence of this phase depends on the parameter $\gamma = c$, which can be interpreted as the minimum *rationality* level of the society. For instance, if all agents are sufficiently rational, i.e., c is large, they may converge to $s^{\max}$ in a single iteration. However, the algorithm converges even if agents are boundedly rational. In Phase 2 of the algorithm, the agents stop updating their contributions and only updates the gradients d_i and the center accumulates them to update w^t.

Remark 5. The convergence rate is $\mathcal{O}(\ln {}^1\!/\!_\epsilon)$ which is same as that of Algorithm 1. However, Algorithm 2 is cleaner in terms of the phases of convergence, the choices of the step sizes γ, η, and yields a guarantee to converge to the desired NE $(w^{\mathtt{OPT}}, s^{\max})$.

Thanks to the smoothness and strong convexity properties of $f(\cdot, s^{\max})$, we can show that running Algorithm 2 guarantees that the model iterate w^T converges to $w^{\mathtt{OPT}}$ at an exponential speed as well.

Corollary 1 (Iterate Convergence)**.** *With the same setup as above and $\gamma = c$ (universal constant), $\eta = \frac{2}{M+\nu}$, we obtain*

$$\|w^T - w^{\mathtt{OPT}}\|_2 < \epsilon, \;\; \textit{and} \;\; s^T = s^{\max},$$

where $T = \kappa + \tilde{T}_0$, with the same κ as in Equation (17) and $\tilde{T}_0 > \left(\ln \frac{\|w^0 - w^{\mathtt{OPT}}\|_2}{\epsilon} \right) / \left(\ln \frac{1 + \frac{\nu}{M}}{1 - \frac{\nu}{M}} \right)$.

Proof. The proof of this follows in the same lines as Theorem 2. Using the choice of β, we first ensure that with κ steps, we obtain $s_i^\kappa = s_i^{\max}$ for all $i \in N$. Now, the framework is same as minimization of a strongly convex and smooth function $f(w)$ with initialization $w^\kappa = w^0$. Using Wright and Recht (2022), with $\eta = \frac{2}{M+\nu}$, we obtain the iterate convergence, namely $\|w^{\tilde{T}_0} - w^{\texttt{OPT}}\|_2 \leq \left(\frac{1-\frac{\nu}{M}}{1+\frac{\nu}{M}}\right)^{\tilde{T}_0} \|w^0 - w^{\texttt{OPT}}\|_2$. Taking log both sides implies the result. □

5 Experiments

Our results in Sections 3 and 4 guarantee convergence of Algorithms 1 and 2 to a pure Nash equilibrium. However, it uses bounds that may be weaker than the actual convergence of the algorithms. It is therefore important to ask how quickly these FL algorithms converge in practice with real datasets.

When the dataset is *non-homogeneously* split across the agents, another important question is to quantify the amount of payment and utilities each agent gets in Algorithm 2. We address all these questions through experiments in this section.

In the first experiment, we arbitrarily sample data from the MNIST dataset (Lecun *et al.*, 1998) and assign 200 distinct data points to all agents, i.e., $s_i^{\max} = 200, \forall i \in N$. We assume the accuracy function to be given by

$$a_i(w, s) = r_i - b_i \cdot \frac{L_i(w, s)}{\|s\|_1},$$

where $r_i = 1$, $b_i = 1$, L_i is a loss function, and $\|\cdot\|_1$ is the L_1-norm. The agents are training a fully connected neural network with one hidden layer (Simard *et al.*, 2003) and the loss function $L_i(w, s)$ is the cross-entropy loss of that network when the model parameter is w and the agents choose the contribution vector s. The accuracy function captures the dependence of both the parameter vector w and the strategy vector s. The gradient of the accuracy function shared with the center captures the gradient of the loss function scaled by the inverse of the total *chosen* data-size of all agents. This scaling factor represents the fact that higher contribution from the agents imply a better accuracy for all agents and is also empirically observed in Kaplan *et al.* (2020). The constant terms r_i and b_i are agent specific and affect the accuracies linearly.

For every agent i, we consider a linear cost $c_i \cdot s_i$, where $c_i = 0.005, \forall i \in N$. We choose β for `2P-UPBReD` to be 1. Learning rates for `UPBReD` are $\gamma = 0.5$, $\eta = 0.5$, and for `2P-UPBReD`, we choose $\gamma = 10$, $\eta = 0.5$. We run both algorithms for an error margin of $\epsilon = 0.01$. Figure 2 shows the results. The convergence time plots were generated by running the experiment 40 times with different (w^0, s^0) for every number of agents. All experiments are run on 48 Intel(R) Xeon(R) Platinum 8168 CPUs with 24 cores and 2 threads per core. The total number of parameters in the model used were 50890, with the hidden layer having 64 neurons.

In the second experiment, we consider a non-homogeneous data distribution among the agents. The accuracy function parameters are $r_i = 3, b_i = 1$, cost gradients of each agent is $c_i = 0.005$, and $\beta = 0.01$. The dataset of size 2750 is distributed as $50, 100, 150, \ldots, 500$ among $n = 10$ agents. The utilities and payments in `2P-UPBReD` is calculated and shown in Figure 3. This plot shows that `2P-UPBReD` rewards data contributors with payments taken from the data consumers, even though both types of agents get positive utilities.

6 Conclusions and future work

In this paper, we proposed `UPBReD` that ensures convergence to a Nash equilibrium and simultaneous convergence of the model parameters w. However, such a model parameter can

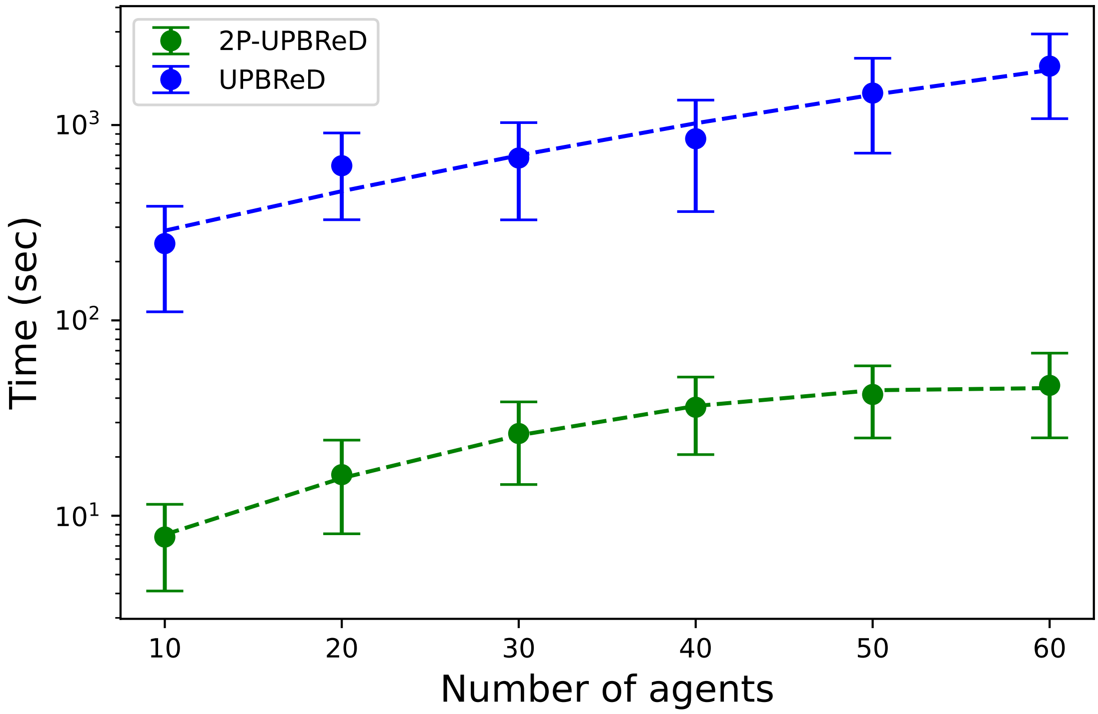


Figure 2: *The y-axis shows the convergence times of* `UPBReD` *and* `2P-UPBReD`*, where dashed lines denote the quadratic fit of the average times for that number of agents.*

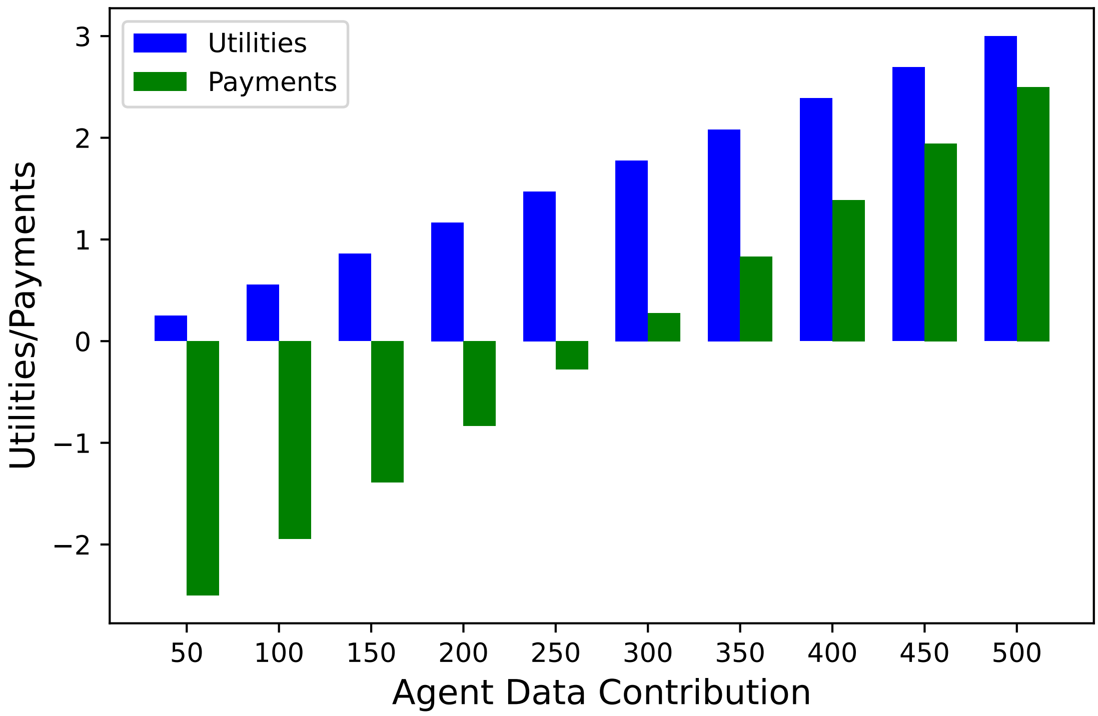


Figure 3: *The x-axis of the plot shows the different $s_i^{\max}$ of the agents $i \in N$, and the y-axis shows the payments and utilities received by them in* `2P-UPBReD` *(negative values denote payment made* by *the agent).*

be off from the optimal $w^{\texttt{OPT}}$, and we proposed an updated two-phase mechanism 2P-UPBReD that ensures full contribution from the agents and convergence to $w^{\texttt{OPT}}$. The trade-off is that in the second case, we need to use monetary transfer, though, these transfers can be only internal, i.e., within the agents.

An immediate future work of this paper is to consider a maximum budget constraint for each agent. The mechanism can only charge that amount of payment at max and ensure some approximation to the properties that we provide here. Also, in realistic situations, all agents do not sample their data from the same distribution. Hence an extension to this work where data owners can cluster and learn the optimal parameters for their cluster will be another important future work.

References

Avrim Blum, Nika Haghtalab, Richard Lanas Phillips, and Han Shao. One for one, or all for all: Equilibria and optimality of collaboration in federated learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1005–1014. PMLR, 18–24 Jul 2021. (Cited on pages 2 and 3)

Bhaskar Ray Chaudhury, Aniket Murhekar, Zhuowen Yuan, Bo Li, Ruta Mehta, and Ariel D. Procaccia. Fair federated learning via the proportional veto core. In *Forty-first International Conference on Machine Learning*, 2024. (Cited on page 3)

Yae Jee Cho, Divyansh Jhunjhunwala, Tian Li, Virginia Smith, and Gauri Joshi. To federate or not to federate: Incentivizing client participation in federated learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022. (Cited on page 2)

Kate Donahue and Jon Kleinberg. Model-sharing games: Analyzing federated learning under voluntary participation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):5303–5311, May 2021. (Cited on pages 2 and 3)

Drew Fudenberg. *Game theory*. MIT press, 1991. (Cited on page 5)

Liang Gao, Li Li, Yingwen Chen, Wenli Zheng, ChengZhong Xu, and Ming Xu. Fifl: A fair incentive mechanism for federated learning. In *Proceedings of the 50th International Conference on Parallel Processing*, ICPP '21, New York, NY, USA, 2021. Association for Computing Machinery. (Cited on page 3)

Eirini Georgoulaki and Kostas Kollias. Arbitrary profit sharing in federated learning utility games. In Argyrios Deligkas and Aris Filos-Ratsikas, editors, *Algorithmic Game Theory*, pages 58–70, Cham, 2023. Springer Nature Switzerland. (Cited on page 3)

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020. (Cited on page 2)

Avishek Ghosh, Raj Kumar Maity, Swanand Kadhe, Arya Mazumdar, and Kannan Ramachandran. Communication efficient and byzantine tolerant distributed learning. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2545–2550. IEEE, 2020. (Cited on page 2)

Avishek Ghosh, Raj Kumar Maity, Swanand Kadhe, Arya Mazumdar, and Kannan Ramchandran. Communication-efficient and byzantine-robust distributed learning with error feedback. *IEEE Journal on Selected Areas in Information Theory*, 2(3):942–953, 2021. (Cited on page 2)

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021. (Cited on page 1)

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. (Cited on page 15)

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pages 3252–3261. PMLR, 2019. (Cited on page 2)

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. *arXiv preprint arXiv:2006.09365*, 2020. (Cited on page 2)

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020. (Cited on pages 2 and 12)

Sai Praneeth Karimireddy, Wenshuo Guo, and Michael I Jordan. Mechanisms that incentivize data sharing in federated learning. *arXiv preprint arXiv:2207.04557*, 2022. (Cited on pages 1 and 2)

Jakub Konečnỳ. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. (Cited on pages 1 and 2)

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. (Cited on page 15)

Xiao-Bai Li and Srinivasan Raghunathan. Pricing and disseminating customer data with privacy awareness. *Decision Support Systems*, 59:63–73, 2014. (Cited on pages 10 and 12)

Wuxing Mao, Qian Ma, Guocheng Liao, and Xu Chen. Game analysis and incentive mechanism design for differentially private cross-silo federated learning. *IEEE Transactions on Mobile Computing*, 23(10):9337–9351, 2024. (Cited on page 3)

Michael Maschler, Shmuel Zamir, and Eilon Solan. *Game theory*. Cambridge University Press, 2020. (Cited on page 5)

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. (Cited on pages 1, 2, and 5)

Aniket Murhekar, Zhuowen Yuan, Bhaskar Ray Chaudhury, Bo Li, and Ruta Mehta. Incentives in federated learning: Equilibria, dynamics, and mechanisms for welfare maximization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. (Cited on pages 1, 2, 3, 7, and 12)

John F Nash Jr. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950. (Cited on page 5)

Shashi Raj Pandey, Lam Nguyen, and Petar Popovski. Fedtoken: Tokenized incentives for data contribution in federated learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022. (Cited on page 3)

Malcolm Pemberton and Nicholas Rau. *Mathematics for economists: an introductory textbook*. University of Toronto Press, 2011. (Cited on page 12)

Bhaskar Ray Chaudhury, Linyi Li, Mintong Kang, Bo Li, and Ruta Mehta. Fairness in federated learning via core-stability. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5738–5750. Curran Associates, Inc., 2022. (Cited on page 3)

Patrice Y. Simard, Dave Steinkraus, and John Platt. Best practices for convolutional neural networks applied to visual document analysis. Institute of Electrical and Electronics Engineers, Inc., August 2003. (Cited on page 15)

Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. *Advances in neural information processing systems*, 31, 2018. (Cited on page 2)

Xiaoli Tang and Han Yu. Competitive-cooperative multi-agent reinforcement learning for auction-based federated learning. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4262–4270. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track. (Cited on page 3)

Xiaoli Tang and Han Yu. Multi-session multi-objective budget optimization for auction-based federated learning. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2024. (Cited on page 3)

Nurbek Tastan, Samar Fares, Toluwani Aremu, Samuel Horváth, and Karthik Nandakumar. Redefining contributions: Shapley-driven federated learning. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 5009–5017. International Joint Conferences on Artificial Intelligence Organization, 8 2024. Main Track. (Cited on page 2)

Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. Ldp-fed: Federated learning with local differential privacy. In *Proceedings of the third ACM international workshop on edge systems, analytics and networking*, pages 61–66, 2020. (Cited on page 2)

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, 15:3454–3469, 2020. (Cited on page 2)

Stephen J. Wright and Benjamin Recht. *Optimization for Data Analysis*. Cambridge University Press, 2022. (Cited on pages 14 and 15)

Yunchao Yang, Yipeng Zhou, Miao Hu, Di Wu, and Quan Z. Sheng. Bara: Efficient incentive mechanism with online reward budget allocation in cross-silo federated learning. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4478–4485. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track. (Cited on page 3)

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pages 5650–5659. Pmlr, 2018. (Cited on pages 2 and 12)

Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A fairness-aware incentive scheme for federated learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, page 393–399, New York, NY, USA, 2020. Association for Computing Machinery. (Cited on page 3)

Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021. (Cited on page 1)