

DualGFL: Federated Learning with a Dual-Level Coalition-Auction Game

Xiaobing Chen¹, Xiangwei Zhou¹, Songyang Zhang², Mingxuan Sun³

¹Division of Electrical and Computer Engineering, Louisiana State University

²Department of Electrical and Computer Engineering, University of Louisiana at Lafayette

³Division of Computer Science and Engineering, Louisiana State University

{xchen87, xwzhou}@lsu.edu, songyang.zhang@louisiana.edu, msun@csc.lsu.edu

Abstract

Despite some promising results in federated learning using game-theoretical methods, most existing studies mainly employ a one-level game in either a cooperative or competitive environment, failing to capture the complex dynamics among participants in practice. To address this issue, we propose **DualGFL**, a novel **Federated Learning** framework with a **Dual-level Game** in cooperative-competitive environments. DualGFL includes a lower-level hedonic game where clients form coalitions and an upper-level multi-attribute auction game where coalitions bid for training participation. At the lower-level DualGFL, we introduce a new auction-aware utility function and propose a Pareto-optimal partitioning algorithm to find a Pareto-optimal partition based on clients' preference profiles. At the upper-level DualGFL, we formulate a multi-attribute auction game with resource constraints and derive equilibrium bids to maximize coalitions' winning probabilities and profits. A greedy algorithm is proposed to maximize the utility of the central server. Extensive experiments on real-world datasets demonstrate DualGFL's effectiveness in improving both server utility and client utility.

Introduction

Federated learning enables decentralized model training across edge devices without transmitting raw data to a central server, which reduces resource costs during local training and model transmission while maintaining performance. The conventional FedAvg approach (McMahan et al. 2017) reduces communication overhead by scheduling a subset of clients per training round. To address data and system heterogeneity and improve efficiency or performance (Chen et al. 2024a; Jing et al. 2024), existing studies have explored various techniques, such as statistical client selection (Chen, Horvath, and Richtarik 2022; Zeng, Yan, and Zhang 2021; Cao et al. 2022; Nguyen et al. 2021), heuristic client selection (Lai et al. 2021; Li et al. 2022), and optimization-based approaches (Wang et al. 2019; Luo et al. 2021; Chen et al. 2024b).

In practical deployments, clients and servers have distinct objectives: clients seek access to the latest global model and potential monetary payoffs (Yang et al. 2023). In contrast, servers aim to incentivize high-quality data contributions

within budget limits (Zhang et al. 2023a; Zhang, Shen, and Bai 2023). Robust incentive mechanisms are therefore essential, and game theory offers a promising framework by modeling strategic interactions among rational agents, improving utilities (Arisdakessian et al. 2023; Wu et al. 2023) and enhancing overall system efficiency (Charatsaris, Diamanti, and Papavassiliou 2023; Zhang et al. 2023b).

Existing game-theoretical approaches in federated learning often use one-level games, either in a purely cooperative (Arisdakessian et al. 2023; Charatsaris, Diamanti, and Papavassiliou 2023) or competitive (Thi Le et al. 2021; Wu et al. 2023; Mai et al. 2022; Hu et al. 2023; Lim et al. 2020) setting, missing complex dynamics among participants in practice. Additionally, current methods tend to improve either client utility (Arisdakessian et al. 2023; Charatsaris, Diamanti, and Papavassiliou 2023) or server utility (Zhang et al. 2023a; Yang et al. 2023; Zhang et al. 2023b; Wang et al. 2022), rarely addressing both simultaneously. This narrow focus limits their applicability in real-world, large-scale federated learning scenarios, where both cooperative and competitive dynamics are present, and the goals of the server and clients must be balanced.

Given these limitations, we pose the question: *Can we design a comprehensive game-theoretical framework for practical federated learning that benefits both the server's and clients' utilities?* To address the question above, we propose an innovative **Federated Learning** framework with a **Dual-level Game** in cooperative-competitive environments, namely **DualGFL**, which meets the following criteria:

1. Clients can autonomously decide whether to participate in the training based on their utilities.
2. The server can autonomously select participants to optimize its utility.

To achieve this goal, we leverage the hierarchical structure in hierarchical federated learning (HFL) to develop a dual-level game-theoretical framework with both cooperative and competitive elements. In HFL, edge servers act as intermediaries between clients and the central server, aggregating model updates from clients and transmitting the aggregated model to the central server. This hierarchical structure helps reduce communication overhead (Liu et al. 2020), manage system heterogeneity (Abad et al. 2020; Briggs, Fan, and Andras 2020), and enhance scalability (Wang et al. 2023).

The main contributions can be summarized as follows:

(1) DualGFL is the first game-theoretical federated learning framework to implement a dual-level game structure. At the lower level, a coalition formation game allows clients to autonomously assess and choose edge servers to form coalitions based on their preferences and utilities. At the upper level, a competitive multi-attribute auction game enables these coalitions to bid for participation in the training process, with the central server determining the winners to maximize its utility.

(2) Unlike existing single-level games, we introduce a new utility function for clients that considers both cooperative and competitive dynamics. This auction-aware utility function evaluates not only the payoff and cost in the lower-level game but also the expected competitiveness in the upper-level auction game.

(3) For efficient and dynamic coalition formation, we propose a Pareto-optimal partitioning (POP) algorithm to find a Pareto-optimal partition based on clients' preference profiles. POP leverages the central server for coalition formation, eliminating the need for client-to-client information exchange and achieving $O(KN^3)$ complexity.

(4) We propose a multi-attribute auction with resource constraints for the upper-level game and provide the theoretical analysis of the equilibrium bid for each coalition.

Additionally, we formulate a score maximization problem for the central server and propose a greedy algorithm with a score-to-resource ratio to solve the coalition selection.

(5) Through experiments on real-world datasets, we evaluate the performance of DualGFL on the server's and clients' utilities. Results demonstrate that DualGFL effectively improves the clients' averaged utility and significantly increases the server's utility while achieving better test accuracy than baselines with single-level games.

Related Work

Incentive mechanisms, modeled through game theory, motivate participants to engage and contribute high-quality data. Existing approaches include the following.

Client-oriented methods mainly involve coalition formation games where clients dynamically change coalitions to maximize utility, such as increasing profits or reducing training costs. For instance, A coalitional federated learning scheme has been proposed where clients form coalitions via a client-to-client trust establishment mechanism (Arisdakessian et al. 2023). A joint optimization problem of the user-to-edge-server association and the uplink power allocation is formulated in HFL, and a satisfaction equilibrium is found to solve the partition problem (Charatsaris, Diamanti, and Papavassiliou 2023). While effective in improving client utility, these methods suffer from high communication overhead and privacy concerns due to peer-to-peer exchanges. *Our DualGFL reduces communication overhead by utilizing a central server for coalition formation, eliminating the need for extensive client-to-client exchanges.*

Server-oriented methods aim to improve the server's utility by incentivizing clients' data contribution and selecting high-quality clients. Auction games are used for client

selection in (Thi Le et al. 2021; Wu et al. 2023). Additional studies explore double auction mechanisms in (Mai et al. 2022), multi-player simultaneous games in (Hu et al. 2023), and private reward games in (Zhang et al. 2023a) to incentivize data contributions, along with evolutionary games for cooperation and latency minimization in (Yang et al. 2023; Zhang et al. 2023b). Regarding games in HFL, mechanisms like MaxQ (Li, Du, and Chen 2022) introduce matching games to assign clients to edge servers based on mutual preferences, and blockchain-based incentives in (Wang et al. 2022) provide public authority and fairness. However, these methods focus on server utility without adequately balancing client incentives. *Our DualGFL addresses this problem by incorporating a dual-level game that simultaneously optimizes the utilities of both clients and the server.*

Mixed methods are designed to serve the utilities of both clients and the server, which is important for practical implementations. A Stackelberg game-based multifactor incentive mechanism for federated learning (SGMFIFL) (Chen et al. 2023) is proposed to incentivize clients to provide more data at low costs and allow clients to change their training strategies adaptively. Similarly, a Stackelberg game and deep reinforcement learning are used to find the optimal strategies for both clients and the server (Zhan et al. 2020). Existing work has shown improved results in terms of clients' and the server's utility. However, compared with server-oriented methods, mixed methods are much less explored. Moreover, even in HFL, existing work only adopts single-level games in either competitive or cooperative environments, failing to model the complex interactions of participants in practice. *In our work, DualGFL integrates both cooperative and competitive dynamics through a dual-level game, enhancing the hierarchical federated learning framework by comprehensively addressing both server utility and client utility.*

System Model

We introduce HFL (preliminary) followed by the cost model.

Hierarchical Federated Learning

In an HFL system, one central server, K edge servers, and N clients collaboratively train a model. The central server aggregates global models, while edge servers aggregate local models from clients. Client i owns a private dataset $\mathcal{D}_i = \{\xi_j^i \mid j = 1, 2, \dots, |\mathcal{D}_i|\}$ of size $|\mathcal{D}_i|$, with $\mathcal{D} = \bigcup_{i \in \mathcal{N}} \mathcal{D}_i$ representing the overall data. The local loss function on dataset \mathcal{D}_i is defined as $F_i(x) := \frac{1}{|\mathcal{D}_i|} \sum_{\xi_j^i \in \mathcal{D}_i} F_i(x; \xi_j^i)$, where x represents the model parameters. HFL systems aim to find an optimal model parameter x to minimize the global loss function $f(x): \min_x f(x) := \sum_{i=1}^N d_i F_i(x)$, where $d_i = |\mathcal{D}_i|/|\mathcal{D}|$ represents the proportion of data held by client i , with $\sum_{i=1}^N d_i = 1$.

In HFL, edge servers relay model updates between clients and the central server. Assume there are T global rounds needed for model convergence. In each round, the training process includes model broadcasting, local training, model transmission, and aggregation. The central server randomly selects clients to participate in the t -th round, forming the participant set $\mathcal{M}^{(t)}$, and broadcasts the current global

model parameters x_t to edge servers. Edge servers relay parameters to selected clients who perform local training over I iterations and send updated model parameters back to edge servers. Edge servers aggregate local models and transmit them to the central server, where global aggregation is performed: $x_{t+1} = \sum_{i \in \mathcal{M}^{(t)}} d_i y_{t,I}^i$, where $y_{t,I}^i$ are the updated model parameters from client i after I local iterations.

Cost Model

Model training in HFL involves both computation and communication costs for clients. The computation cost is associated with local data processing, while the communication cost involves transmitting model updates to the edge server. Formally, we define the computation cost as $c_{i,cp} = \kappa a_i f_i^2$, where κ is a coefficient based on the CPU architecture, a_i is the number of CPU cycles, and f_i is the CPU clock frequency (Zhang et al. 2018).

The communication cost depends on the coalition S that client i is in. In this work, we consider a wireless environment as an exemplary application with the achievable uplink transmission rate $r_i(S) = B \log_2(1 + p_i h_i / N_0)$, where B denotes the bandwidth, p_i and h_i are the transmission power and channel gain between client i and its edge server, and N_0 denotes the noise power spectral density (Wu et al. 2023). Hence, the communication cost is $c_{i,cm}(S) = \frac{|x|}{r_i(S)}$, where $|x|$ denotes the size of model updates. Therefore, the total cost of client i is given by

$$C_i(S, \theta_i) = c_{i,cp} + \theta_i c_{i,cm}(S), \quad (1)$$

where θ_i denotes the cost factor of client i . Note that although we define C_i in the wireless scenario, our proposed DualGFL can be generalized to any networking system by customizing C_i for specific applications.

Dual-Level Game Federated Learning

Architecture

DualGFL consists of three hierarchies: clients, edge servers, and the central server, and involves two-level games: a lower-level coalition formation game and an upper-level auction game. DualGFL includes four main components: coalition formation, bidding, coalition selection, and federated training and payoff distribution, as shown in Figure 1. In each training round, the following steps are executed. (1) **Coalition Formation**: Clients calculate utilities of potential coalitions and form preference profiles. These profiles are submitted to the central server who finds a *Pareto-optimal* (Aziz, Brandt, and Harrenstein 2013) partition that cannot be improved to make one client better off without making at least one client worse off. Each coalition contains one edge server and several clients. (2) **Bidding**: Edge servers place bids in the upper-level auction game to participate in the training. (3) **Coalition Selection**: The central server selects coalitions based on received bids. The winning bids are determined by the predefined rules, and every client publicly knows these rules in advance. (4) **Federated Training and Payoff Distribution**: Selected coalitions receive the global model, perform local training, and send aggregated updates

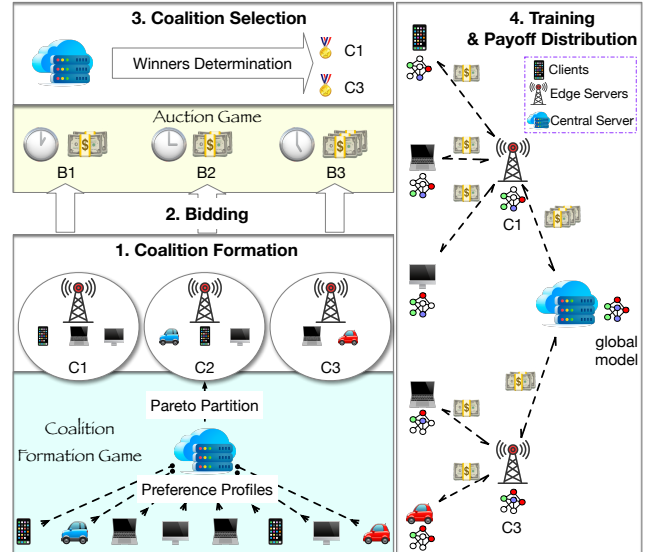


Figure 1: DualGFL architecture.

to the central server. The central server then distributes payoffs to coalitions, which are further distributed to individual members by the edge servers.

Lower-Level Coalition Formation Game

We introduce the hedonic game for coalition formation, propose an auction-aware utility function, and propose the POP algorithm to find a Pareto-optimal partition.

Hedonic Game We formulate the lower-level coalition formation game as a hedonic game where clients use a utility function to evaluate their coalition preferences.

Definition 1. A *hedonic game* G is a pair $(\mathcal{N}, \mathcal{R})$, where \mathcal{N} is a finite set of players and $\mathcal{R} = \{\mathcal{R}_i : i \in \mathcal{N}\}$ denotes the preference profile of all the players. Here \mathcal{R}_i denotes the preference profile of player i and is defined by a binary, complete, reflexive, and transitive preference relation \succeq_i on the set of coalitions that player i belongs to, i.e., $\mathcal{N}_i = \{S \subseteq 2^{\mathcal{N}} : i \in S\}$. The strict and indifference preference relations are denoted by \succ_i and \sim_i , respectively.

Remark: In our setting, each valid coalition must contain **one and only one** edge server. We also assume **individual rationality**, meaning each client is at least as well off in a coalition as they would be alone.

Clients aim to maximize their personal utility when choosing a coalition. The utility function of client i in coalition S is:

$$U_i(S) = R_i(S) - C_i(S, \theta_i), \quad (2)$$

where $R_i(S)$ denotes the payoff from the coalition and $C_i(S, \theta_i)$ is the training cost as defined in Equation (1).

Auction-Aware Preference Profile Generation Unlike existing work on the design of the utility function for single-level games (Arisdakessian et al. 2023), we propose a novel auction-aware utility function, incorporating both lower-level game payoffs and upper-level auction competitiveness.

Based on this utility, we introduce an auction-aware preference profile generation algorithm. Specifically, we first calculate the auction-aware payoff $R_i(S)$, which can be calculated by

$$R_i(S) = w_i(S) \cdot \mathbb{E}(R^{(t)}(S)), \quad (3)$$

where $\mathbb{E}(R^{(t)}(S))$ is the expected total payoff for coalition S and $w_i(S) = \frac{d_i}{\sum_{j \in S} d_j}$ represents the data contribution proportion. $\mathbb{E}(R^{(t)}(S))$ is given by

$$\mathbb{E}(R^{(t)}(S)) = Pr(S \in \mathcal{M}^{(t)}) \cdot R^{(t)}(S), \quad (4)$$

where $Pr(S \in \mathcal{M}^{(t)})$ is the probability of coalition S being selected and $R^{(t)}(S)$ denotes the true payoff.

Since other coalitions' bids are unknown (Che 1993), we estimate $\mathbb{E}(R^{(t)}(S))$ using historical earnings by exponential moving average (EMA):

$$\hat{R}^{(t)}(S) = \begin{cases} \alpha_e \hat{R}^{(t-1)}(S) + (1 - \alpha_e) R^{(t)}(S), & \text{if } S \in \mathcal{M}^{(t)}, \\ \hat{R}^{(t-1)}(S), & \text{otherwise,} \end{cases} \quad (5)$$

where α_e is the EMA coefficient. Then, the auction-aware payoff $R_i(S)$ and auction-aware utility $U_i(S)$ are

$$R_i(S) = \frac{d_i \hat{R}^{(t)}(S)}{\sum_{j \in S} d_j} \quad (6)$$

and

$$U_i(S) = \frac{d_i \hat{R}^{(t)}(S)}{\sum_{j \in S} d_j} - C_i(S, \theta_i), \quad (7)$$

respectively. The preference function $P_i(S)$, based on utility $U_i(S)$, is

$$P_i(S) = \begin{cases} U_i(S), & \text{if } S \notin h(i), \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where $h(i)$ contains the coalitions that client i has joined when the client requests to join a new partition. This function reduces candidate coalitions, lowering computational complexity (Saad et al. 2011). Clients generate preference profiles \mathcal{R}_i by comparing coalitions using $P_i(S)$, resulting in: $S \succeq_i T \Leftrightarrow P_i(S) \geq P_i(T)$. To reduce the computation complexity and without loss of generality, clients in Dual-GFL must use \succ_i and \sim_i only in their preference profiles.

Pareto-Optimal Partitioning Algorithm The lower-level hedonic game aims to find a Pareto-optimal partition π^* of disjoint coalitions. We propose the POP algorithm to achieve the goal. We define a partition as follows.

Definition 2. A *partition* $\pi = \{S_1, S_2, \dots, S_K\}$ of \mathcal{N} consists of K disjoint coalitions, where $S_k \neq \emptyset$, $\bigcup_{k=1}^K |S_k| = N$, and $S_k \cap S_l = \emptyset$ for $k \neq l$. Let $\pi(i) = \{S \in \pi : i \in S\}$ be the coalition player i is in. Let $\prod(\mathcal{N})$ denote the collection of all coalition structures in \mathcal{N} .

Pareto dominance is defined as follows.

Definition 3. A partition π *Pareto dominates* π' if $\pi \succeq_i \pi'$ for any player i and if $\pi \succ_j \pi'$ for at least one player j . A partition π is *Pareto-optimal* if no partition π' Pareto dominates it.

Algorithm 1: Pareto-Optimal Partitioning (POP)

Input: Preference profiles $\mathcal{R} = \{\mathcal{R}_i : i \in \mathcal{N}\}$

Output: Pareto-optimal partition π^*

```

1 Initialize  $\mathcal{R}^\top = \mathcal{R}$ ;
2 Initialize  $\mathcal{R}^\perp$  by replacing all  $\succ$  in  $\mathcal{R}$  with  $\sim$ ;
3  $\pi^* = \text{PerfectPartition}(\mathcal{N}, \mathcal{R}^\perp)$ ;
4 for  $i \in \mathcal{N}$  do
5   while  $\mathcal{R}_i^\perp \neq \mathcal{R}_i^\top$  do
6      $\mathcal{R}'_i = \text{Refine}(\mathcal{R}_i^\perp, \mathcal{R}_i^\top)$ ;
7      $\mathcal{R}' = (\mathcal{R}_1^\perp, \dots, \mathcal{R}_{i-1}^\perp, \mathcal{R}'_i, \mathcal{R}_{i+1}^\perp, \dots, \mathcal{R}_n^\perp)$ ;
8      $\pi = \text{PerfectPartition}(\mathcal{N}, \mathcal{R}')$ ;
9     if  $\pi \neq \emptyset$  then
10       $\pi^* = \pi$ ;
11       $\mathcal{R}_i^\perp = \mathcal{R}'_i$ ;
12     else
13       $\mathcal{R}_i^\top = \mathcal{R}_i^\perp$ ;

```

Definition 4. Given a preference profile \mathcal{R} , a partition π is *perfect* if for any player i , $\pi(i)$ is the most preferred coalition, i.e., $\pi(i) \succeq_i S$ for any $S \in \mathcal{N}_i$.

Perfect partitions, the best for all players, usually do not exist in federated learning. However, Pareto optimality can be obtained by finding the perfect partition on relaxed preference profile (Aziz, Brandt, and Harrenstein 2013).

We first present the following theorem that characterizes the relationship between the perfect partition and the Pareto-optimal partition. The detailed proof of Theorem 1 can be found in (Aziz, Brandt, and Harrenstein 2013).

Theorem 1. (Pareto-Optimal Partition) Let $(\mathcal{N}, \mathcal{R}^\top)$ and $(\mathcal{N}, \mathcal{R}^\perp)$ represent hedonic games where $\mathcal{R}^\perp \leq \mathcal{R}^\top$, which means that \mathcal{R}^\top has some preferences that are more strict than the ones in \mathcal{R}^\perp . Suppose π is a perfect partition for \mathcal{R}^\perp . Then π is Pareto-optimal for \mathcal{R}^\top if and only if there exists a preference profile $\mathcal{R} \in [\mathcal{R}^\perp, \mathcal{R}^\top]$ such that

1. π is perfect for \mathcal{R} , and
2. no partition is perfect for any \mathcal{R}' where $\mathcal{R} < \mathcal{R}' \leq \mathcal{R}^\top$.

Based on Theorem 1 and preference refinement in (Aziz, Brandt, and Harrenstein 2013), we propose the Pareto-optimal partitioning (POP) algorithm that accepts clients' preference profiles and assigns clients to edge servers to form a Pareto-optimal partition, as shown in Algorithm 1.

Initially, POP sets \mathcal{R}^\top to be the original preference profiles and \mathcal{R}^\perp to be the relaxed version where all \succ are replaced with \sim . The initial perfect partition π^* is computed using the *PerfectPartition* function on \mathcal{R}^\perp , which is guaranteed to exist. The *PerfectPartition* function matches the mutual preferences of edge servers and clients by shuffling preferences and iterating through preferred clients, adding them to coalitions if the edge server is the client's top choice.

POP then iteratively refines each player's preference profile from \mathcal{R}^\perp towards \mathcal{R}^\top . The *Refine* function gradually changes indifferences to strict preferences, and *PerfectPartition* seeks a perfect partition for the current profile. If found, the partition is updated. This iterative process ensures the final partition π^* is Pareto-optimal for \mathcal{R}^\top .

Refine and PerfectPartition Functions

Function Refine ($\mathcal{R}_i^\perp, \mathcal{R}_i^\top$):
 | Replace one \sim in \mathcal{R}_i^\perp with \succ towards \mathcal{R}_i^\top ;
Function PerfectPartition (\mathcal{N}, \mathcal{R}):
 | Initialize π with empty lists for each edge server;
 | Shuffle edge server preferences and search order;
for each edge server do
 | **for each client in server's preference list do**
 | **if server is client's top choice then**
 | | Add client to server's coalition;
if all clients are allocated then
 | **return** π
else
 | **return** \emptyset

Discussion of POP: POP returns a Pareto-optimal partition. Since initial π^* always exists and there is no partition perfect for \mathcal{R}' where $\mathcal{R} < \mathcal{R}' \leq \mathcal{R}^\top$ when POP terminates. Theorem 1 ensures that POP returns a Pareto-optimal partition. All the Pareto-optimal partitions can be found by changing the iteration order in line 4 of Algorithm 1. For computational efficiency, we only use the first one found.

POP algorithm runs in polynomial time. The computational complexity of POP is dominated by its iterative process and the *PerfectPartition* function. With N clients and K edge servers, the *PerfectPartition* function operates with $O(KN)$ complexity due to nested loops. The outer loop runs N times, and each iteration of the while loop could run up to $(N-1)$ times, leading to an overall complexity of $O(KN^3)$.

Upper-Level Multi-Attribute Auction with Resource Constraints

We introduce the multi-attribute auction model, derive equilibrium bids to maximize coalition utility, and propose an algorithm to solve the coalition selection problem.

Multi-Attribute Auction Model We formulate the competition among coalitions for training participation as a procurement auction, in which the central server is the buyer and K coalitions are the suppliers. The auction proceeds as follows: 1. Each supplier independently places a multi-dimensional bid containing price, qualities, and resource requests. 2. The buyer calculates a score for each bid and selects M winners based on the scores under the resource budget. 3. Winning bids are formalized into contracts, and winners must deliver according to their bids.

Specifically, the bid of coalition k is formulated as $B_k = (P_k, \mathbf{Q}_k, E_k)$, where $P_k \in \mathbb{R}_+$ denotes the price, $\mathbf{Q}_k = (q_1^k, q_2^k, \dots, q_l^k) \in \mathbb{R}_+^l$ denotes the quality vector where each entry represents a nonmonetary attribute, such as data volume, data quality, and delivery time. E_k denotes the resource request and we use communication bandwidth as the resource in this paper.

The score for each bid is calculated using a scoring rule $h : \mathbb{R}^{l+1} \rightarrow \mathbb{R}$, designed to reflect the buyer's preferences. We use a quasi-linear scoring rule:

$$h(P_k, \mathbf{Q}_k) = \mathbf{Q}_k^T \boldsymbol{\alpha} - P_k, \quad (9)$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_l) \in \mathbb{R}_+^l$ are quality weights. The buyer aims to select winners maximizing the total score, while suppliers aim to maximize profits by optimal bids.

Equilibrium Bids of Coalitions In the auction, each supplier's profit is the difference between the price and the cost if its bid wins. The utility of a supplier is

$$\pi_k(P_k, \mathbf{Q}_k) = \begin{cases} P_k - C_k(\mathbf{Q}_k, \theta_k), & \text{if } x_k = 1, \\ 0, & \text{if } x_k = 0, \end{cases} \quad (10)$$

where $C_k(\mathbf{Q}_k, \theta_k)$ denotes the total cost to provide qualities \mathbf{Q}_k with private cost factor θ_k and $x_k \in \{0, 1\}$ is the winning indicator. The total cost $C_k(\mathbf{Q}_k, \theta_k)$ is the sum of members' costs: $C_k(\mathbf{Q}_k, \theta_k) = \sum_{i \in S} C_i(S, \theta_i)$.

Suppliers aim to maximize profit by optimizing bids. Consider that supplier S is one of the winners in the auction with a score ψ_k to fulfill under the scoring rule $h(\cdot)$. We can formulate the utility maximization problem for the supplier as

$$\max_{P_k, \mathbf{Q}_k} \pi_k(P_k, \mathbf{Q}_k) \quad (P1)$$

$$\text{s.t. } h(P_k, \mathbf{Q}_k) = \psi_k. \quad (P1a)$$

This can be converted to an unconstrained problem:

$$\max_{\mathbf{Q}_k} \mathbf{Q}_k^T \boldsymbol{\alpha} - C_k(\mathbf{Q}_k, \theta_k) - \psi_k. \quad (P2)$$

Based on the scoring auction theory (Che 1993), we give the solution to Problem P2 in Theorem 2.

Theorem 2. (Equilibrium Bid) *The multi-attribute auction with a quasi-linear scoring rule has a unique and symmetric equilibrium bid for each supplier, given by*

$$\begin{aligned} \mathbf{Q}_k^*(\theta_k) &= \arg \max (\mathbf{Q}_k^T \boldsymbol{\alpha} - C_k(\mathbf{Q}_k, \theta_k)), \\ P_k^*(\theta_k) &= C_k(\mathbf{Q}_k^*, \theta_k) + \tilde{P}(\mathbf{Q}_k^*, \theta_k). \end{aligned} \quad (11)$$

Here, \tilde{P} denotes the profit calculated by

$$\tilde{P}(\mathbf{Q}_k^*, \theta_k) = \int_{\underline{\theta}}^{\bar{\theta}} C_{\theta}(\mathbf{Q}_k^*(t), t) \left[\frac{1 - F(t)}{1 - F(\theta_k)} \right]^{N-1} dt,$$

where θ_k is independently and identically distributed over $[\underline{\theta}, \bar{\theta}]$ and $F(\cdot)$ is the cumulative distribution function.

The optimal qualities $\mathbf{Q}_k^*(\theta_k)$ are computed based on the scoring rule, independent of the fulfillment score ψ_k . The optimal price includes cost and additional profit based on optimal qualities and cost factor distribution.

Score Maximization Problem The buyer aims to select winners who can maximize the total score, that is:

$$\begin{aligned} \max_{\{x_k\}} & \sum_{k=1}^K x_k (\mathbf{Q}_k^T \boldsymbol{\alpha} - P_k) \\ \text{s.t.} & \sum_{k=1}^K x_k = M, \quad \sum_{k=1}^K x_k E_k \leq E_{max}, \\ & x_k = \{0, 1\}, \forall k \in [1, K], \end{aligned} \quad (P3)$$

where M denotes the number of winners and E_{max} denotes the bandwidth budget.

Dataset	Method	Total Score	Avg. Client Quality	Avg. Coalition Quality	Avg. Client Utility	Test Accuracy
FMNIST (0.6)	FedAvg	-	1198.19 (1x)	1198.19 (1x)	125.92 (1x)	91.14 %
	FedAvgAuc	-	1581.94 (1.32x)	1581.94 (1.32x)	206.59 (1.64x)	89.54%
	FedAvgHed	13.97 (1x)	1179.91 (0.98x)	6702.52 (5.59x)	225.01 (1.79x)	91.15%
	DualGFLStat	343.84 (24.60x)	1209.09 (1.01x)	7833.43 (6.54x)	272.71 (2.17x)	90.93%
	DualGFL	544.87 (38.99x)*	1271.56 (1.06x)	12470.54 (10.41x)	483.06 (3.84x)	91.36%
FMNIST (0.1)	FedAvg	-	1201.88 (1x)	1201.88 (1x)	109.77 (1x)	88.26%
	FedAvgAuc	-	2208.00 (1.84x)	2208.00 (1.84x)	255.21 (2.33x)	88.08%
	FedAvgHed	10.21 (1x)	1200.10 (1.00x)	6814.46 (5.67x)	217.62 (1.98x)	86.46%
	DualGFLStat	331.71 (32.50x)	1225.99 (1.02x)	7917.18 (6.59x)	261.28 (2.38x)	88.12%
	DualGFL	516.29 (50.58x)	1331.40 (1.11x)	12473.63 (10.38x)	447.59 (4.08x)	88.43%
EMNIST (0.1)	FedAvg	-	697.08 (1x)	697.08 (1x)	4.43 (1x)	82.86%
	FedAvgAuc	-	1392.56 (2.00x)	1392.56 (2.00x)	16.42 (3.71x)	74.92%
	FedAvgHed	325.95 (1x)	698.88 (1.00x)	6968.68 (10.00x)	28.01 (6.33x)	83.22%
	DualGFLStat	615.96 (1.89x)	702.11 (1.01x)	8062.36 (11.57x)	33.36 (7.54x)	82.67%
	DualGFL	938.72 (2.88x)	820.74 (1.18x)	12293.74 (17.64x)	55.38 (12.51x)	83.72%
CIFAR10 (0.1)	FedAvg	-	995.67 (1x)	995.67 (1x)	77.14 (1x)	73.50%
	FedAvgAuc	-	1857.13 (1.87x)	1857.13 (1.87x)	188.29 (2.44x)	73.93%
	FedAvgHed	238.49 (1x)	983.21 (0.99x)	5513.51 (5.54x)	180.18 (2.34x)	66.30%
	DualGFLStat	293.33 (1.23x)	1044.23 (1.05x)	6761.98 (6.79x)	230.57 (2.99x)	73.17%
	DualGFL	488.18 (2.05x)	1161.65 (1.17x)	11240.82 (11.29x)	420.56 (5.45x)	75.17%

Table 1: Performance comparison between DualGFL and baselines. The best performance in each data configuration is in **bold**. “Total Score” does not apply to FedAvg and FedAvgAuc. Except for the metric “Total Score”, FedAvg is used as the benchmark. For the metric “Total Score”, FedAvgHed is the benchmark. The improvement ratio in parentheses, such as (38.99x), denotes the improvement with respect to the benchmark. Our DualGFL significantly improves server utility and client utility.

Problem P3 is a 0-1 Knapsack problem with a cardinality and a resource constraint, which is NP-hard. We propose a greedy algorithm with a score-to-resource ratio to solve the problem. The algorithm involves two steps: 1. Calculate the score-to-resource ratio for each supplier: $\frac{Q_k^T \alpha - P_k}{E_k}$, and sort suppliers in descending order. 2. Select winners from the sorted list until M suppliers are chosen or E_{max} is reached. The algorithm’s computational complexity is dominated by the sorting step, resulting in efficient $O(K \log K)$ complexity, making it suitable for large-scale applications.

Experiments

Datasets and Predictive Model: We use the FMNIST (Xiao, Rasul, and Vollgraf 2017), EMNIST (Cohen et al. 2017), and CIFAR10 (Krizhevsky and Hinton 2009) datasets for image classification tasks. We implement a shallow convolution neural network with two convolution layers as the classification model in (Panchal et al. 2023).

Data Heterogeneity: To simulate real-world data in HFL, which is none independently and identically distributed (non-I.I.D.), we use Dirichlet data partitioning (Panchal et al. 2023) to partition original datasets into clients’ private datasets. We set multiple data configurations: FMNIST (0.1), FMNIST (0.6), EMNIST (0.1), and CIFAR10 (0.1), where values in parenthesis denote the Dirichlet parameters.

Network Simulations: We randomly generate graphs to simulate the network topology. Edge servers are placed in a grid with 100 km intervals, and clients are randomly placed within this grid. The maximum coalition size is constrained by a hyperparameter $|S|_{max}$.

Metrics: We evaluate DualGFL using: **Test Accuracy:**

Prediction accuracy on test data. **Total Score:** Total score of winning coalitions. **Average Coalition Quality:** Average quality of winning coalitions. **Average Client Quality:** Average quality of winning clients. **Average Client Utility:** Average utility clients gain from participation. Higher values in each metric represent better performance. Results are averaged over three runs with different random seeds.

Baselines: We compare our DualGFL with the following baselines: **FedAvg** (McMahan et al. 2017): The number of selected clients is adjusted to approximate the number of winning clients in DualGFL. **FedAvg Auction (FedAvgAuc)** (Thi Le et al. 2021): Only the auction game is applied to FedAvg, where clients place bids directly without forming coalitions. **FedAvg Hedonic (FedAvgHed)** (Arisdakessian et al. 2023): Only the hedonic game is applied to FedAvg, where clients form coalitions, and the central server randomly selects coalitions. **DualGFL Statistics (DualGFLStat):** A variant of DualGFL where winning coalitions are randomly selected according to their normalized scores.

System Parameters: For FMNIST (0.6), FMNIST (0.1), and CIFAR10 (0.1), we generate $N = 50$ clients and $K = 9$ edge servers, selecting $M = 3$ coalitions in each round. The maximum coalition size is $|S|_{max} = 10$. For EMNIST (0.1), we generate $N = 1000$ clients and $K = 100$ edge servers, selecting $M = 5$ coalitions in each round. The maximum coalition size is $|S|_{max} = 15$. Each experiment is conducted in $T = 250$ rounds, and clients update the model for $I = 3$ epochs using the SGD optimizer with a learning rate of 0.01 and momentum of 0.9. The batch size is set to 32. The central server uses data size as the quality metric.

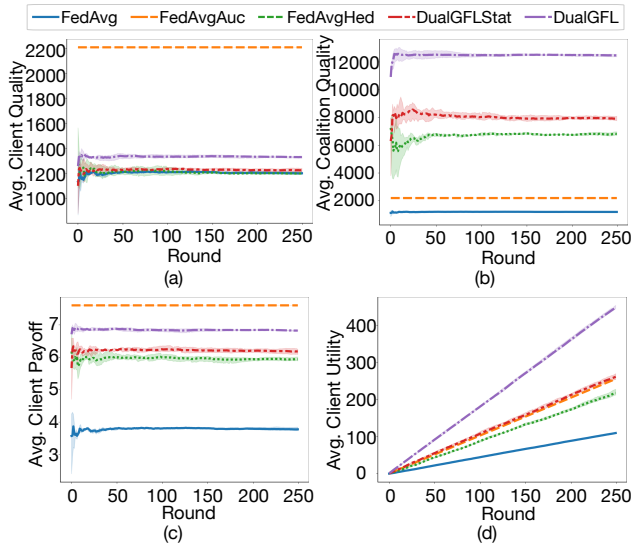


Figure 2: Training dynamics of key metrics. (a), (b), (c), and (d) show the cumulative average of client quality, coalition quality, client payoff, and client utility, respectively.

Experiment Results

DualGFL shows superior performance in server utility, including total score, average client quality, and average coalition quality. As shown in Table 1, DualGFL achieves improvements of at least 2.05 times in total score, 1.06 times in average client quality, and 10.38 times in average coalition quality. DualGFLStat shows improvement over FedAvgHed, indicating that score-based selection is superior to random selection for high-quality participants.

DualGFL significantly outperforms baselines in client utility. DualGFL provides the highest average client utility, achieving improvement up to 12.51 times over FedAvg. This significant improvement demonstrates DualGFL’s effectiveness in enhancing client welfare. DualGFLStat achieves the second-best performance, suggesting that clients benefit more from participating in the dual-level game of DualGFL than single-level game methods.

DualGFL outperforms the baselines in test accuracy across all settings. Although accuracy improvement is not the primary objective, DualGFL shows strong accuracy performance, especially in the CIFAR10 (0.1) setting with high data heterogeneity, achieving approximately a 1.7% improvement in accuracy due to the selection of higher-quality coalitions and clients.

Evaluating Training Dynamics DualGFL significantly improves server and client utility, as shown by key metrics obtained at the end of the training in Table 1. To understand the performance improvement, we present training dynamics for key metrics in the FMNIST (0.1) setting in Figure 2. Specifically, Figures 2(a), 2(b), 2(c), and 2(d) show the cumulative average of client quality, coalition quality, client payoff, and client utility during the training, respectively. Shaded areas indicate variations across three runs.

DualGFL effectively enhances participant quality.

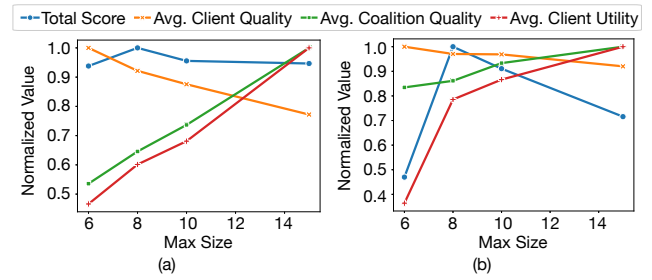


Figure 3: Impact of max size $|S|_{max}$ on the performance of the coalition selection. (a) and (b) show the performance of DualGFL and DualGFLStat, respectively.

There are distinct tiers in average client quality and coalition quality from Figure 2(a) and 2(b), with FedAvgAuc and DualGFL consistently selecting better quality clients. Despite having the highest client quality, FedAvgAuc shows low coalition quality due to the lack of coalition formation.

Clients gain higher payoffs by joining higher-quality coalitions. The average client payoff stabilizes as training progresses, as shown in Figure 2(c), with consistent differences among methods, which explains the stable and linear increase in client utility as shown in Figure 2(d). The distribution of client payoff mirrors coalition quality except for FedAvgAuc, indicating clients benefit from participating in high-quality coalitions.

Ablation Study: Increasing the maximum coalition size $|S|_{max}$ affects the performance of DualGFL. We conduct an ablation study on $|S|_{max}$ in FMNIST (0.1) setting. Smaller values produce more uniformly sized coalitions. We tested $|S|_{max} \in [6, 8, 10, 15]$. Figure 3 shows normalized values for the total score, average client quality, average coalition quality, and average client utility across different $|S|_{max}$.

Larger $|S|_{max}$ produces higher average coalition quality but lower client quality. Larger $|S|_{max}$ increases size discrepancy among coalitions. Coalitions with size advantage are prone to win the auction, leading to higher average coalition quality. However, larger coalitions attract more “free riders” with lower quality, causing a decrease in average client quality but still increasing average client utility. The total score peaks at $|S|_{max} = 8$, balancing client movement among coalitions and avoiding the Matthew effect, where larger coalitions have a dominant advantage.

Conclusion

We introduced DualGFL, the first federated learning framework combining a dual-level game to enhance both server and client utilities. The lower level uses a hedonic game for coalition formation, where we proposed an auction-aware utility function and a Pareto-optimal partitioning algorithm. The upper level formulates a multi-attribute auction game with resource constraints, deriving equilibrium bids and solving a score maximization problem for the central server. Experiments show that DualGFL consistently outperforms single-game baseline models across various metrics.

Acknowledgments

This work was supported in part by the NSF under Grant No. 1943486, No. 2246757, No. 2315612, and No. 2332011, and in part by a grant from BoRSF under contract LEQSF(2024-27)-RD-B-03.

References

- Abad, M. S. H.; Ozfatura, E.; GUndUz, D.; and Ercetin, O. 2020. Hierarchical Federated Learning ACROSS Heterogeneous Cellular Networks. In *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*, 8866–8870.
- Arisdakessian, S.; Wahab, O. A.; Mourad, A.; and Otrók, H. 2023. Coalitional Federated Learning: Improving Communication and Training on Non-IID Data With Selfish Clients. *IEEE Trans. Serv. Comput.*, 16(4): 2462–2476.
- Aziz, H.; Brandt, F.; and Harrenstein, P. 2013. Pareto optimality in coalition formation. *Games and Economic Behavior*, 82: 562–581.
- Briggs, C.; Fan, Z.; and Andras, P. 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 1–9.
- Cao, H.; Pan, Q.; Zhu, Y.; and Liu, J. 2022. Birds of a Feather Help: Context-aware Client Selection for Federated Learning. In *Proc. Int. Workshop Trustable, Verifiable and Auditable Federated Learning with AAAI*.
- Charatsaris, P.; Diamanti, M.; and Papavassiliou, S. 2023. On the Accuracy-Energy Tradeoff for Hierarchical Federated Learning via Satisfaction Equilibrium. In *Int. Conf. Distrib. Comput. Smart Syst. Internet Things (DCOSS-IoT)*, 422–428. Los Alamitos, CA, USA: IEEE Computer Society.
- Che, Y.-K. 1993. Design Competition Through Multidimensional Auctions. *RAND J. ECON*, 24(4): 668–680.
- Chen, W.; Horvath, S.; and Richtarik, P. 2022. Optimal client sampling for federated learning. *Trans. Machine Learning Research (TMLR)*.
- Chen, X.; Zhou, X.; Zhang, H.; Sun, M.; and Vincent Poor, H. 2024a. Client Selection for Wireless Federated Learning With Data and Latency Heterogeneity. *IEEE Internet Things J.*, 11(19): 32183–32196.
- Chen, X.; Zhou, X.; Zhang, H.; Sun, M.; and Zhao, T. 2024b. Cost-Effective Federated Learning: A Unified Approach to Device and Training Scheduling. In *Proc. IEEE Int. Conf. Commun. (ICC)*, 3488–3493.
- Chen, Y.; Zhou, H.; Li, T.; Li, J.; and Zhou, H. 2023. Multi-factor Incentive Mechanism for Federated Learning in IoT: A Stackelberg Game Approach. *IEEE Internet Things J.*, 10(24): 21595–21606.
- Cohen, G.; Afshar, S.; Tapson, J.; and Van Schaik, A. 2017. EMNIST: Extending MNIST to handwritten letters. In *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 2921–2926.
- Hu, Q.; Wang, S.; Xiong, Z.; and Cheng, X. 2023. Nothing Wasted: Full Contribution Enforcement in Federated Edge Learning. *IEEE Trans. Mob. Comput.*, 22(05): 2850–2861.
- Jing, S.; Yu, A.; Zhang, S.; and Zhang, S. 2024. FedSC: Provable Federated Self-supervised Learning with Spectral Contrastive Objective over Non-i.i.d. Data. In *Proc. Int. Conf. Machine Learning (ICML)*, 22304–22325.
- Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario.
- Lai, F.; Zhu, X.; Madhyastha, H. V.; and Chowdhury, M. 2021. Oort: Efficient Federated Learning via Guided Participant Selection. In *Proc. USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 19–35.
- Li, C.; Zeng, X.; Zhang, M.; and Cao, Z. 2022. PyramidFL: A Fine-Grained Client Selection Framework for Efficient Federated Learning. In *Proc. Annu. Int. Conf. Mob. Comput. Netw. (MobiCom)*, 158–171. ISBN 9781450391818.
- Li, Z.; Du, H.; and Chen, X. 2022. A Two-Stage Incentive Mechanism Design for Quality Optimization of Hierarchical Federated Learning. *IEEE Access*, 10: 132752–132762.
- Lim, W. Y. B.; Xiong, Z.; Miao, C.; Niyato, D.; Yang, Q.; Leung, C.; and Poor, H. V. 2020. Hierarchical Incentive Mechanism Design for Federated Machine Learning in Mobile Networks. *IEEE Internet Things J.*, 7(10): 9575–9588.
- Liu, L.; Zhang, J.; Song, S.; and Letaief, K. B. 2020. Client-Edge-Cloud Hierarchical Federated Learning. In *Proc. IEEE Int. Conf. Commun. (ICC)*, 1–6.
- Luo, B.; Li, X.; Wang, S.; Huang, J.; and Tassiulas, L. 2021. Cost-Effective Federated Learning in Mobile Edge Networks. *IEEE J. Sel. Areas Commun. (JSAC)*, 39(12): 3606–3621.
- Mai, T.; Yao, H.; Xu, J.; Zhang, N.; Liu, Q.; and Guo, S. 2022. Automatic Double-Auction Mechanism for Federated Learning Service Market in Internet of Things. *IEEE Trans. Netw. Sci. Eng.*, 9(5): 3123–3135.
- McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 1273–1282.
- Nguyen, H. T.; Schwag, V.; Hosseinalipour, S.; Brinton, C. G.; Chiang, M.; and Vincent Poor, H. 2021. Fast-Convergent Federated Learning. *IEEE J. Sel. Areas Commun. (JSAC)*, 39(1): 201–218.
- Panchal, K.; Choudhary, S.; Parikh, N.; Zhang, L.; and Guan, H. 2023. Flow: Per-instance Personalized Federated Learning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, volume 36, 18712–18755. Curran Associates, Inc.
- Saad, W.; Han, Z.; Basar, T.; Debbah, M.; and Hjørungnes, A. 2011. Hedonic Coalition Formation for Distributed Task Allocation among Wireless Agents. *IEEE Trans. Mob. Comput.*, 10(9): 1327–1344.
- Thi Le, T. H.; Tran, N. H.; Tun, Y. K.; Nguyen, M. N. H.; Pandey, S. R.; Han, Z.; and Hong, C. S. 2021. An Incentive Mechanism for Federated Learning in Wireless Cellular Networks: An Auction Approach. *IEEE Trans. Wireless Commun. (TWC)*, 20(8): 4874–4887.

Wang, S.; Tuor, T.; Salomidis, T.; Leung, K. K.; Makaya, C.; He, T.; and Chan, K. 2019. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE J. Sel. Areas Commun. (JSAC)*, 37(6): 1205–1221.

Wang, X.; Zhao, Y.; Qiu, C.; Liu, Z.; Nie, J.; and Leung, V. C. M. 2022. InFEDge: A Blockchain-Based Incentive Mechanism in Hierarchical Federated Learning for End-Edge-Cloud Communications. *IEEE J. Sel. Areas Commun. (JSAC)*, 40(12): 3325–3342.

Wang, Z.; Xu, H.; Liu, J.; Xu, Y.; Huang, H.; and Zhao, Y. 2023. Accelerating Federated Learning With Cluster Construction and Hierarchical Aggregation. *IEEE Trans. Mob. Comput.*, 22(7): 3805–3822.

Wu, C.; Zhu, Y.; Zhang, R.; Chen, Y.; Wang, F.; and Cui, S. 2023. FedAB: Truthful Federated Learning With Auction-Based Combinatorial Multi-Armed Bandit. *IEEE Internet Things J.*, 10(17): 15159–15170.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*.

Yang, P.; Zhang, H.; Gao, F.; Xu, Y.; and Jin, Z. 2023. Multi-player evolutionary game of federated learning incentive mechanism based on system dynamics. *Neurocomputing*, 557: 126739.

Zeng, X.; Yan, M.; and Zhang, M. 2021. Mercury: Efficient On-Device Distributed DNN Training via Stochastic Importance Sampling. In *Proc. ACM Conf. Embedded Networked Sensor Systems*, 29–41. ISBN 9781450390972.

Zhan, Y.; Li, P.; Qu, Z.; Zeng, D.; and Guo, S. 2020. A Learning-Based Incentive Mechanism for Federated Learning. *IEEE Internet Things J.*, 7(7): 6360–6368.

Zhang, C.; Shen, T.; and Bai, F. 2023. Toward Secure Data Sharing for the IoT Devices With Limited Resources: A Smart Contract-Based Quality-Driven Incentive Mechanism. *IEEE Internet Things J.*, 10(14): 12012–12024.

Zhang, J.; Hu, X.; Ning, Z.; Ngai, E. C.-H.; Zhou, L.; Wei, J.; Cheng, J.; and Hu, B. 2018. Energy-Latency Tradeoff for Energy-Aware Offloading in Mobile Edge Computing Networks. *IEEE Internet Things J.*, 5(4): 2633–2645.

Zhang, L.; Zhu, T.; Xiong, P.; Zhou, W.; and Yu, P. S. 2023a. A Game-Theoretic Federated Learning Framework for Data Quality Improvement. *IEEE Trans. Knowl. Data Eng.*, 35(11): 10952–10966.

Zhang, S.; Xu, Y.; Liu, J.; Takakura, H.; Chen, L.; and Shitoro, N. 2023b. Bandwidth Allocation for Low-Latency Wireless Federated Learning: An Evolutionary Game Approach. In *Proc. IEEE Int. Conf. Commun. (ICC)*, 1628–1633.