Adaptive Incentivize for Federated Learning with Cloud-Edge Collaboration under Multi-level Information Sharing

Shijing Yuan, *Member,IEEE* Beiyu Dong, Jie Li, *Fellow IEEE*, Song Guo, *Fellow, IEEE*, Hongyang Chen, *Senior Member IEEE*, Chentao Wu, *Member IEEE*, Jie Wu, *Fellow IEEE*, Wei Zhao, *Fellow, IEEE*,

Abstract—Federated Learning with Cloud-Edge Collaboration (FL-CEC) has emerged as a cutting-edge paradigm in distributed learning. Efficient resource investment incentive mechanisms are crucial to encouraging clients in FL-CEC to contribute the necessary data and computational resources for training. However, existing studies are inadequate in meeting the incentive design requirements under multi-level information-sharing scenarios. Moreover, current works often rely on specific functional relationships between resource investment and global model accuracy. To bridge these gaps, this paper investigates the incentive problem for data and computational resource investment under multi-level information-sharing levels. We design a resource investment incentive mechanism based on a weighted potential game without depending on any specific functional relationship between data investment and model accuracy. Furthermore, we propose four algorithms to solve resource investment strategies for different levels of information sharing. The complexity and convergence rates of the proposed algorithms are thoroughly analyzed. Finally, we construct a simulation incentive platform on Aliyun. Extensive evaluations demonstrate that the proposed scheme effectively enhances social welfare, and improves collaborative training accuracy and efficiency.

Index Terms—FL-CEC, adaptive incentivize, potential game, stochastic learning, policy gradient method.

1 Introduction

With the emergence of the Cloud-Edge Collaboration (CEC) architecture, distributed learning under a multi-dimensional CEC framework has gradually become a prominent research area. Among these, Federated Learning with CEC (FL-CEC) has evolved into a cutting-edge distributed learning paradigm [2]. FL-CEC has shown promising application prospects in various fields, such as the Industrial Internet of Things (IIoT), biomedicine, and the financial industry [3], [4]. In FL-CEC, edge nodes collect data from end devices and determine the strategies for data and computational resource input to participate in the local training process of the model. The cloud server acts as the coordination center, responsible for distributing model parameters to the edge nodes and aggregating the model updates obtained from the local training at the edge nodes, thereby realizing iterative optimization of the global model [5]. However, the lack of appropriate incentive mechanisms may lead to selfish behavior among devices in FL-CEC, as they might be unwilling to invest the required data and computational resources due to concerns about resource consumption [6].

Efficient incentives for resource input are crucial to encouraging FL-CEC participants to contribute the necessary data and computational resources actively. However, designing efficient resource input incentive mechanisms faces two key challenges. First, the level of information sharing among edge devices varies, ranging from full to no sharing. For example, clients may only share limited decision variable information (such as resource input strategies), or they may make independent decisions about resource input in a distributed environment without sharing any decision variables or parameter information. This variation increases the difficulty of modeling and solving for reasonable resource input strategies [4]. Second, due to network fluctuations and dynamic changes in the participants' available computational and data resources, the overhead of collaborative training may vary dynamically.

Existing research on FL-CEC incentives has focused on customer input evaluation [7], [8], fairness [9], personalized incentives [10], privacy overhead compensation [11], [12], hitchhike prevention, and social welfare maximization [13]–[16]. However, current studies are insufficient to address the incentive design requirements under multi-level information-sharing challenges. Most existing work assumes that training participants will disclose sensitive information, such as training costs and unit model performance gains, to evaluate better and incentivize resource inputs and formulate strategies to maximize individual benefits. Yet, in real FL-CEC scenarios, participants may be reluctant to disclose such information due to privacy concerns. In addition, existing research lacks com-

This research was supported by fundings from the Hong Kong RGC General Research Fund (152244/21E, 152169/22E, 152228/23E, 162161/24E), Research Impact Fund (No. R5011-23F, No. R5060-19), Collaborative Research Fund (No. C1042-23GF), NSFC/RGC Collaborative Research Scheme (No. CRS_HKUST602/24), Areas of Excellence Scheme (No. AoE/E-601/22-R), and the InnoHK (HKGAI), in part by the National Natural Science Foundation of China under Grant 62271452, and the Research and Development Program of China Telecom under Grants T-2025-27.

S. Yuan is with the Department of Fundamental Network Operations, China Telecom Research Institute, Shanghai, China. E-mail:dr.sj.yuan@ieee.org. B. Dong, J. Li and C. Wu are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. E-mail: {sato_kenji_dby, lijiecs, wct}@sjtu.edu.cn. (Corresponding authors: Jie Li and Song Guo).

S. Guo is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, E-mail:songguo@cse.ust.hk.*

H. Chen is with the Zhejiang Lab, Hangzhou 311121, China. Email: dr.h.chen@ieee.org.

J. Wu is with the Department of Computer and Information Sciences, Temple University, USA. E-mail: jiewu@temple.edu.

Wei Zhao is with CAS Shenzhen Institute of Advanced Technology, Shenzhen 518055, China. E-mail: zhao.wei@siat.ac.cn.

This is an extended and enhanced version of the paper [1] that has been accepted for presentation at the 43rd IEEE ICDCS, July 2023.

prehensive consideration of the actual relationship between resource input and model accuracy. Most existing work relies on optimization methods and assumes an explicit relationship between specific resource inputs and global model accuracy. However, since model accuracy changes dynamically with the training process, the actual form of this relationship is often unknown. Therefore, existing incentive designs may not accurately reflect the actual resource contributions of training participants.

To bridge these gaps, we investigate the resource input incentive problem under multi-level information sharing. Considering the advantages of game theory in characterizing multi-agent interactions, we leverage potential game theory [17] to design a resource input incentive mechanism based on a weighted potential game[1]. This mechanism aims to encourage edge devices to contribute reasonable data and computational resources, balancing improvements in model accuracy with training overhead, thereby enhancing social welfare.

Technically, we propose four algorithms to solve resource input strategies following different levels of information sharing and resource dynamics. Under global information sharing, we introduce a Centralized algorithm based on Generalized Benders Decomposition (CGBD) to derive near-optimal resource input strategies. In scenarios where only resource input strategy information is shared, we propose a distributed algorithm based on Dynamic Best Response (DBR) for weighted potential games to achieve resource input strategies under Nash equilibrium (NE), a stable state in which no player can improve their utility by unilaterally changing their strategy. Under no information sharing and dynamically changing available resources, we propose a distributed algorithm based on Multi-Agent Stochastic Learning (MASL), which uses an adaptive search of the strategy vector to learn near-optimal resource input strategies. Furthermore, in no-information-sharing scenarios and aiming to ensure strict convergence rate guarantees in non-stationary environments, we propose a Multi-Agent Decentralized Policy Gradient (MAD-PG) algorithm that integrates value function evaluation and strategy function updates to improve the stability of resource input strategies. Finally, we deploy a simulation incentive platform using Aliyun Elastic Compute Service (ECS).

The main contributions are summarized as follows:

- Through pre-experiments, we explore the potential relationship between data input and the accuracy of the trained global model. We propose an incentive mechanism based on a weighted potential game and construct a multi-variable potential function to characterize the interactions among training participants, without assuming any specific functional relationship between data input and model accuracy.
- The proposed CGBD algorithm ensures theoretically near-optimal resource investment strategies through centralized potential function optimization. In contrast, the DBR algorithm achieves low-complexity strategy solutions via distributed best response, requiring only decision information sharing[2]. Additionally, the MASL and MAD-PG algorithms leverage stochastic learning and policy gradients, respectively, to address strategy optimization in complex scenarios with no information sharing and dynamic environmental changes. Theoretical analyses validate the complexity and convergence of the proposed algorithms.
- We develop an incentive platform using Aliyun ECS, enabling clients to decide their resource inputs autonomously. Moreover, we design and implement a smart contract prototype on the platform to realize automated and credible incentive delivery.
- Extensive experimental results based on the real platform and datasets demonstrate that, compared to existing baselines, the proposed scheme effectively improves overall social welfare and enhances collaborative training accuracy and efficiency.

Paper Organization. Section 2 summarizes relevant work. Section 3 describes the system model. Section 4 presents our designed incentive mechanism. Section 5 proposes four algorithms suitable for three levels of information sharing. Section 6 describes the designed platform and evaluates the experiments. Section 7 concludes this paper.

2 Related work

In the realm of collaborative distributed training in CEC, scholars have applied contract theory, game theory, and auction theory to design adaptive incentive mechanisms. For instance, in the edge-end cooperative federated learning scenario, Le et al. [18] and Yuan et al. [19] employed auction theory to maximize social welfare and minimize energy costs, respectively, by converting the incentives between edge servers and mobile devices into auction problems. Deng et al. [20] proposed an auction-based quality-aware incentive mechanism to enhance the training efficiency of FL-CEC. Ng et al. [21] adopted evolutionary game theory to model the dynamic selection process of data owners in their engagement with training clients. Similarly, Khan et al. [22] modeled the interaction between the central server and the wireless devices as a two-layer Stackelberg game to incentivize clients to invest resources in training. Lastly, Ding et al. [23] designed an adaptive incentive mechanism based on contract theory to balance model performance and incentive cost. Other mechanisms focused on maximizing social welfare by analyzing client interactions in competitive markets [9], [24]. Furthermore, blockchain and smart contracts have been utilized to provide trusted incentives [25], [26]. For instance, scholars have used blockchain-based reputation records and smart contract-based mechanisms to guarantee trusted and automatic distribution of incentives [10], [13]. There are also several other incentives designed to improve collaborative training by rationally assessing a client's resource contribution [7], [8], building personalized models [27], maximizing social welfare [14]–[16], and compensating for privacy expenses [11], [12]. For example, Yu et al. [9] investigated the impact of authenticity and market competition on incentive design in CEC multiparty machine learning. Additionally, Tang et al. [15] proposed a public good-based incentive mechanism to motivate clients to provide more computational resources but overlooked the amount of local data clients should invest in training. Chen et al. [28]

1. A type of non-cooperative game where a weighted potential function exists, ensuring that the change in individual utility resulting from a player's strategy adjustment can be linearly mapped to the change in the potential function, thereby guaranteeing the existence of a pure strategy Nash equilibrium [1]

2. The source code is available at https://github.com/sato-kenjidby/PFLlib-Framework.

addressed the problem of hitchhiking between clients through rational profit distribution and Multi-player Multi-action Zero Determinant (MMZD) but did not guarantee credible incentives for collaborative training. Furthermore, Zhang et al. [14] considered a client's long-term involvement to determine the optimal data input strategy. However, their mechanism relied on the assumption of a data-accuracy function specific to a model type. In summary, the aforementioned incentive designs either assume a specific functional relationship between data input and model accuracy [14]–[16] or overlook the diversity in the degree of information sharing. Consequently, these incentive mechanisms do not apply to dynamic distributed training scenarios under CEC.

On the other hand, reinforcement learning (RL)-based approaches exhibited potential in distributed training with CEC, such as RL-driven blockchain adaptive sharding mechanisms [29]. Zhang et al. [30] proposed a Multi-Agent Reinforcement Learning (MARL) framework for federated learning, which optimizes processing latency and model accuracy by selecting high-quality clients. Guo et al. [31] presented a centralized RL-based approach that improves the accuracy of training models by adaptively adjusting the size of the subset of participants and the training batch's size. Similarly, Zhang et al. [32] developed a MARL algorithm for FL-CEC that reduces energy consumption and accelerates the convergence of training by efficiently allocating training participants and managing resources. Moreover, Zhan et al. [33] proposed a centralized RL-based approach that instructs clients to adaptively invest computational resources to minimize the total overhead of training, which includes the weighting of training time and energy consumption. In resource-limited systems, Zheng et al. [34] proposed an algorithm based on the deep deterministic policy gradient (DDPG) to balance energy consumption and the accuracy of trained models. In contrast to existing RL-based approaches to improve collaborative training in FL-CEC, we integrate incentive design, automated execution of incentives, and trustworthy guarantees in the context of information sharing across various degrees.

In summary, our work is distinctive because: (1) we consider the dynamics of available resources and training overhead in FL-CEC, as well as the challenges of incentivizing resource contributions under varying levels of information sharing; (2) we construct an innovative incentive mechanism based on a weighted potential game, without relying on any specific data-accuracy function; (3) we propose four algorithms to address the resource input strategy problem under different levels of information sharing. These algorithms solve multi-variable complex potential functions without depending on a specific data-accuracy function relationship.

3 System Model

This section provides an overview of the system, including the payoffs obtained through co-training, the impact of data resource input on accuracy, and the training overhead involved.

3.1 Overview

We consider a FL-CEC framework, illustrated in Fig 1, comprising a cloud server (model aggregation node) and multi-

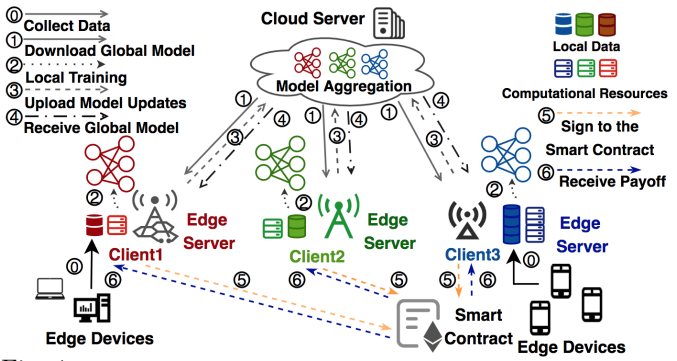


Fig. 1: Overview of FL-CEC, illustrating the interactions among edge servers, i.e., clients, the cloud server, and the incentive mechanism based on smart contracts. The training workflow includes 1) edge servers collecting local data; 2) the cloud server distributing the global model; 3) edge servers training the model using local data; and 4) the trained updates are uploaded and aggregated by the cloud server.

ple edge servers (clients).[3] We denote the set of clients as $\mathcal{O} = \{o_i\}_{i\in\mathcal{N}}$, where $\mathcal{N}$ is the set of clients' identification numbers. Each client possesses computational and data resources. Specifically, let $\mathcal{S}_i$ denote client o_i's local dataset with $|\mathcal{S}_i|$ as its size, and let $F_i^{(m)}$ denote the total computational resource of o_i. To participate in co-training, o_i must contribute at least $D_{min}|\mathcal{S}_i|$ data, where $0 < D_{min} \leq 1$, otherwise the global model will be unavailable to o_i. During co-training, clients decide what proportion of their available data and computational resources to use, to maximize individual revenue. After training, each client o_i signs a smart contract and receives corresponding benefits based on the incentives designed. The detailed mechanism is provided in Section. 4.1 - 4.2.

The co-training process can be divided into several stages, detailed below, and the corresponding time overheads. 1) **Model downloading:** Each client o_i downloads the global model from the model aggregation node, which costs o_i an average time represented by $T_i^{(1)}$. 2) **Local training**: Each client o_i selects a portion of the dataset, i.e., the $d_i|\mathcal{S}_i|$ data volume and the f_i computational resources to perform local training and obtain a locally updated model. Denoting the weight of the global model as $\boldsymbol{w}$, then the local loss function [1] of o_i, denoted as $L_i(\boldsymbol{w})$, can be calculated as $L_i(\boldsymbol{w}) = \frac{1}{|\mathcal{S}_i|}\sum_{x_i^{(k)}\in\mathcal{S}_i} l\left(\boldsymbol{w}, x_i^{(k)}\right)$, where $x_i^{(k)}$ means the k^{th} data sample in $\mathcal{S}_i$, and $l(\boldsymbol{w}, x_i^{(k)})$ is the loss on $\boldsymbol{w}$ and $x_i^{(k)}$. o_i's average time spent on local training can be calculated as $T_i^{(2)}(d_i, f_i) = \frac{\eta_i d_i|\mathcal{S}_i|}{f_i}$, where η_i is the computational intensity, presenting the computational resources required for processing a unit amount of samples. 3) **Model uploading**: Each client o_i uploads the updated local model $\boldsymbol{w}_i$ to the model aggregation node, which costs o_i an average time represented by $T_i^{(3)}$. 4) **Model aggregation**: The model aggregation node updates the global model using the FedAvg algorithm [35], with the loss [1] calculated as

$$L(\boldsymbol{w}) = \sum_{i=1}^{|\mathcal{N}|}\left[\frac{d_i|\mathcal{S}_i|}{\sum_{k=1}^{|\mathcal{N}|} d_k|\mathcal{S}_k|}\right]L_i(\boldsymbol{w}). \quad (1)$$

Therefore, to ensure efficiency, completion of the training process within the designated deadline τ is specified such that

3. As edge servers and cloud servers play crucial roles in FL-CEC, we focus on interactions between the edge nodes and the cloud node in our discussion, without expanding on the end-edge data collection process.

$T_i^{(1)} + T_i^{(2)}(d_i, f_i) + T_i^{(3)} \leq \tau$ holds for each client.

3.2 Training Revenue

Based on the decision d_i for data input, client o_i submits data of amount $d_i|\mathcal{S}_i|$ to obtain a highly accurate global model. Let $\boldsymbol{d_{-i}} = \{d_j\}_{j\neq i, j\in\mathcal{N}}$ denote the tuple of data input strategies for all clients except o_i. The loss of model accuracy for o_i can be expressed as $A(d_i, \boldsymbol{d}_{-i})$, where a lower value of $A(d_i, \boldsymbol{d}_{-i})$ indicates higher model accuracy [14], [15]. Specifically, the *data-accuracy function* $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$[4] can be expressed as the difference between the value of model accuracy loss $A(\mathbf{0})$ when not involved in training and the accuracy loss after inputting data for training [14], [15], i.e., $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) = A(\mathbf{0}) - A(d_i, \boldsymbol{d}_{-i})$. Therefore, the revenue earned by o_i from the global model can be expressed as $p_i\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$, where $p_i > 0$ is defined as o_i's profit factor, i.e., the revenue earned by o_i from a unit improvement in global model performance [36].

In cases where the loss function $L(\boldsymbol{w})$ exhibits strong convexity and all client data is independently and identically distributed (IID), $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ grows as $d_i|\mathcal{S}_i|$ increases, but at a diminishing rate [15]. Building on these observations and prior research [14], [37], we assume that the first and second derivatives of the data-accuracy function

$$\frac{\partial \boldsymbol{P}(d_i, \boldsymbol{d}_{-i})}{\partial d_i} \geq 0, \frac{\partial^2 \boldsymbol{P}(d_i, \boldsymbol{d}_{-i})}{\partial d_i^2} \leq 0. \quad (2)$$

Similarly, when client data is not independently and identically distributed (non-IID), we assume that $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ satisfies $\frac{\partial \boldsymbol{P}(d_i, \boldsymbol{d}_{-i})}{\partial d_i} \geq 0$.

To assess the impact of d_i on $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$, we conduct experiments with classical models such as ResNet-18, AlexNet, Densenet, and MobileNet in the CIFAR-10, FMNIST, SVHN, and EuroSat datasets [38], [39]. Specifically, the "ResNet-CIFAR10" curve highlights the performance improvement achieved with increasing data input in a highly complex visual dataset, with analogous patterns observed for "DenseNet-EuroSat", "AlexNet-FMNIST" and "MobileNet-SVHN". The preliminary results shown in Fig 2- 3 confirm the above assumption.

3.3 Data-Accuracy Function

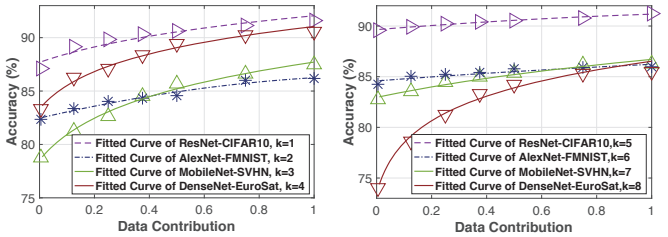
Non-convex function. In scenarios where the client data distribution is uneven or the data quality is unstable, the assumption of convexity for the data-accuracy function $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ is no longer valid. Therefore, we extend $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ to account for the non-convex case, i.e.,

$$\boldsymbol{P}(d_i', \boldsymbol{d}_{-i}) - \boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) \geq 0, \forall d_i' \geq d_i, d_i', d_i \in \mathbb{R}. \quad (3)$$

The properties above suggest that, in most cases of client data distribution, a larger amount of data input by the client leads to higher accuracy of the global model, even when the data distribution is uneven or the data quality is unstable.

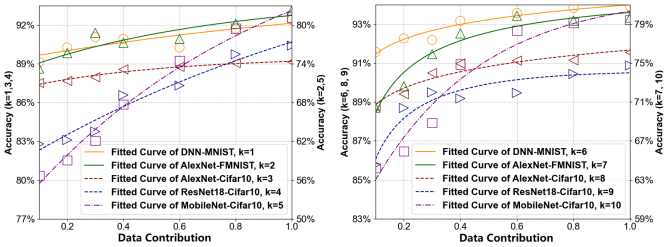
Implicit function. Explicitly capturing the correlation between the amount of data $\{d_i, \boldsymbol{d}_{-i}\}$ invested by the client and the accuracy of the global model $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ may be difficult, particularly in practical co-training environments where the data-accuracy function is often implicit.

4. The global model accuracy is not only dependent on $d_i, \boldsymbol{d}_{-i}$, but also on the number of training iterations and data quality, which are considered as system constants here when analyzing the functional relationship between $d_i, \boldsymbol{d}_{-i}$ and $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ [14].



(a) $|\mathcal{N}| = 2$ (b) $|\mathcal{N}| = 5$

Fig. 2: Relationship between data input and accuracy with the amount of data input $\{|\mathcal{S}_i^k|\}_{i\in\mathcal{N}} \in [2000, 20000]$, and data inputs for all clients except o_i fixed at $d_{i'} = 0.5$, where $d_{i'} \in \boldsymbol{d}_{-i}$.



(a) $|\mathcal{N}| = 2$ (b) $|\mathcal{N}| = 5$

Fig. 3: Data input vs. accuracy under the non-IID setting, where each client holds 3000 samples. For 2-client setups ($k = 1 \sim 5$), Client 1 holds labels $k/2 \sim k$, and Client 0's labels are proportional (e.g., $k = 10$: [0], [0, 1, 2, 3]). For 5-client setups ($k = 6 \sim 10$), Clients 1 ∼4 follow a Dirichlet distribution ($\alpha = 0.5$), and Client 0's labels are proportional.

3.4 Training Overhead

Clients participating in the training process must not only provide data and computational resources but also face various types of training overheads, such as privacy overheads, computational energy overheads, and communication energy overheads [4]. Our focus is on the energy consumption and communication overhead of the training process. The energy consumption overhead can be calculated as $E_i^{comp} = \kappa f_i^2 \eta_i d_i |\mathcal{S}_i|$ [4], [33], where κ represents the effective capacitance of the computing chip, η_i is the energy factor. Additionally, the communication energy consumption can be expressed as $E_i^{comm} = E_{DL}T_i^{(1)} + E_{UL}T_i^{(3)}$, where E_{DL} and E_{UL} denote the communication overhead per unit time of the client during the model download and upload phases, respectively. Thus, the training overhead of client o_i can be calculated as [5]

4 Incentive Mechanism Design

In this section, we present an incentive mechanism based on payoff redistribution and model clients' interactions as a non-cooperated game, specifically, a weighted potential game.

4.1 Mechanism Design

The goal of the mechanism is to encourage each client to participate in training with the optimal resource input strategy and collaboratively enhance the performance of the global model. Based on *payoff redistribution*, the client o_i with less resource input will transfer a portion of its revenue to the client o_j with more resource input according to specific rules.

Definition 1 (Payoff Redistribution)**.** *We define γ as the incentive intensity, representing the price per unit of input resource. The payoff redistribution that client o_i receives from client o_j is denoted as $r_{i,j}$ and can be expressed as follows:*

5. In the FL-CEC, communication overhead is impacted by the wireless access method and network architecture [5]. Meanwhile, training overhead may dynamically change due to network fluctuations and variations in the computational and data resources available to clients.

$$r_{i,j} = \gamma\left(r_i - r_j\right), \tag{4}$$

where $r_i \triangleq d_i|\mathcal{S}_i| + \lambda f_i$ *measures* o_i*'s resource input, with a constant* $\lambda > 0$ *unifying resource units. Thus, the total payoff redistribution received by* o_i *is* $\boldsymbol{R}_i = \sum_{j\in\mathcal{N}} r_{i,j}$.

Under the proposed incentive mechanism, the *payoff* of client o_i integrates the model accuracy revenue, energy overhead, and payoff redistribution, and can be represented as:

$$\begin{aligned}\boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}) &= p_i \boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) - \varpi_e E_i + \boldsymbol{R}_i \\ &= p_i \boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) - \varpi_e \left(\kappa f_i^2 \eta_i d_i |\mathcal{S}_i| + E_i^{(comm)}\right) + \sum_{j\in\mathcal{N}} r_{i,j},\end{aligned} \tag{5}$$

where ϖ_e is the energy consumption coefficient, $\pi_i = \{d_i, f_i\}$ represents the resource input strategy for client o_i, and $\boldsymbol{\pi_{-i}} = \{\pi_j\}_{j\neq i, j\in\mathcal{N}}$ represents the tuple of strategies for all other clients except o_i. Therefore, the *social welfare* of the entire system is $\sum_{i\in\mathcal{N}} \boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i})$.

The mechanism above aims to encourage each client to invest data and computing resources to enhance the performance of the global model while fulfilling individual rationality, computational efficiency, and budget balance. We provide the following definitions for these three properties.

Definition 2 (Individual Rationality)**.** *Individual Rationality (IR) implies that the utility of any client at a Nash equilibrium* $\{\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}\}$ *is non-negative, i.e.,* $\boldsymbol{\mathcal{C}_i}(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}) \geq 0$.

Definition 3 (Computational Efficiency)**.** *Computational Efficiency (CE) indicates that the incentive mechanism should have a manageable computational complexity and be executed efficiently in polynomial time.*

Definition 4 (Budget Balance)**.** *Budget Balance (BB) implies that incentives are self-sustaining without external incentives. That is, the total amount of payoff redistribution is zero, i.e.,* $\sum_{i\in\mathcal{N}} \boldsymbol{R}_i = 0$.

4.2 Prototype Design

To ensure automatic and trustworthy execution of the payoff redistribution between clients without the involvement of a third party, we designed a prototype based on smart contracts, as shown in Fig 4. The smart contract automatically executes the payoff redistribution $r_{i,j}$, thereby preventing malicious behavior of refusing to do so. Additionally, the smart contract records the redistribution results to the blockchain, establishing a trustworthy incentive mechanism. In case of client disputes, these recorded results can be used as a basis for arbitration and enforced.

As illustrated in Fig 4, the smart contract process consists of three steps. Firstly, each o_i is registered into the smart contract and simultaneously makes a margin deposit denominated in Ethereum virtual currency by invoking the $Deposit()$ function[6]. After the training phase, the client o_i submits its optimal resource input configuration $\{d_i^*, f_i^*\}$ to the smart contract by invoking the $ContributeSubmit()$ function[7]. Subsequently, the smart contract computes the payoff through the $CalculatePayoff()$ function, redistributes it through the $PayoffRedistribution()$ function, and returns the margin proportionally. With the proposed mechanism, the proceeds' redistribution is automatically and credibly executed without the involvement of any third party, while the optimal resource input strategy $\{d_i^*, f_i^*\}$ is recorded on the local blockchain by invoking the $ProfileRecord()$ function to ensure the traceability of the payoff redistribution.

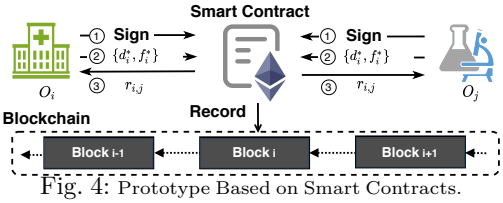


Fig. 4: Prototype Based on Smart Contracts.

4.3 Weighted Potential Game Model

Given its ability to represent interactions among distributed clients, we employ game theory to analyze the optimal resource inputs of the clients. We model the interaction among clients as a non-cooperative game and further prove that it is a weighted potential game.

Each client o_i determines its resource input strategy $\pi_i = \{d_i, f_i\}$ to maximize its payoff, taking into account the other clients' strategies $\boldsymbol{\pi}_{-i}$, which can be formalized as

$$\begin{aligned} \max_{\pi_i} \quad & \boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}) \\ & C_i^{(1)} : d_i \in [D_{min}, 1], \\ & C_i^{(2)} : f_i \in [F_i^{(1)}, \ldots, F_i^{(m)}], \\ & C_i^{(3)} : T_i^{(1)} + T_i^{(2)}(d_i, f_i) + T_i^{(3)} \leq \tau, \end{aligned} \tag{6}$$

where $C_i^{(1)}$ and $C_i^{(2)}$ ensure the feasibility of the resource strategy, and $C_i^{(3)}$ stipulates that the average time o_i spent on training must not exceed a given duration τ.

Then, the game $\boldsymbol{\mathcal{G}}$ can be formalized as $\boldsymbol{\mathcal{G}} = \{\mathcal{O}, \boldsymbol{\pi}, \mathcal{C}\}$, where $\boldsymbol{\pi}$ is the set of strategies for all clients, and $\mathcal{C}$ is the set of utility functions $\{\boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i})\}_{i\in\mathcal{N}}$.

Definition 5 (Nash equilibrium and ϵ-Nash equilibrium)**.** *A strategy profile* $\pi^* = \{\pi_i^*\}_{i\in\mathcal{N}}$ *is the Nash equilibrium (NE) of a game* $\mathcal{G}$ *if no client* o_i *can increase its utility by unilaterally deviating from its strategy, i.e.* $\boldsymbol{\mathcal{C}_i}(\pi_i^*, \boldsymbol{\pi}_{-i}^*) \geq \boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}^*), \forall \pi_i, \forall i \in \mathcal{N}$*, or an* ϵ*-Nash equilibrium* π^* *if no* o_i *can increase its payoff by* ϵ *by unilaterally deviating from its own strategy, i.e.,* $\boldsymbol{\mathcal{C}_i}(\pi_i^*, \boldsymbol{\pi}_{-i}^*) \geq \boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}^*) - \epsilon, \forall \pi_i, \forall i \in \mathcal{N}$.

4.4 Existence of Nash Equilibrium (NE)

Definition 6 (Potential game and potential function)**.** *A potential game is a special type of non-cooperative game, in which the impact of changes in any player's strategy on its payoff can be mapped to the variation of a global value, which thereby defines a potential function [17]. Any set of resource input strategies that locally maximizes the potential function represents a strategy under the NE*[8]*.*

Definition 7 (Weighted potential game)**.** *A game* $\mathcal{G}$ *is classified as a weighted potential game if there exists a potential function* $\boldsymbol{U}(\pi_i, \boldsymbol{\pi}_{-i})$ *that satisfies the following equation for all clients, i.e.,* $z_i \left[\boldsymbol{U}(\pi_i, \boldsymbol{\pi}_{-i}) - \boldsymbol{U}(\pi_i', \boldsymbol{\pi}_{-i})\right] = \boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}) - \boldsymbol{\mathcal{C}_i}(\pi_i', \boldsymbol{\pi}_{-i})$*, where* z_i *represents a positive scaling factor.*

NE strategy always exists in a weighted potential game [17] with a bounded potential function, as the change in

6. Functions defined in the smart contract can be interacted with by external applications in Ethereum through the application binary interface (ABI) [1].

7. The authenticity of $\{d_i^*, f_i^*\}$ submitted by the client can be confirmed by the Trusted Execution Environment (TEE), a penalization mechanism, or past gradient contributions, which are beyond the scope of this work [1].

8. The theory of potential games states that if a properly constructed potential function exists in a non-cooperative game, it can measure the impact of players' strategy changes on the overall system value (potential value) [17]. The process of achieving NE in the game corresponds to the process of the potential function reaching its local optimum.

$\boldsymbol{\mathcal{C}}_i(\pi_i, \boldsymbol{\pi}_{-i})$ resulting from strategy bias for any client o_i is proportional to the change in $\boldsymbol{U}(\pi_i, \boldsymbol{\pi}_{-i})$, with the positive scaling factor z_i.

Theorem 1 (Weighted potential game)**.** *The game $\boldsymbol{\mathcal{G}}$ is a weighted potential game with the potential function given by (7), which guarantees the existence of a pure strategy NE in the game $\boldsymbol{\mathcal{G}}$.*

$$\boldsymbol{U}(\boldsymbol{\pi}) = \boldsymbol{P}\left(d_i, \boldsymbol{d}_{-i}\right) - \sum_{i\in\mathcal{N}}\left[\frac{\varpi_e \kappa f_i^2 \eta_i d_i |\mathcal{S}_i|}{p_i} - \frac{(n-1)\gamma r_i}{p_i}\right]. \tag{7}$$

Proof. Refer to **Appendix** for the detailed proof. □

5 Algorithm Design

In this section, we propose algorithms to address each of the three levels of information sharing. These algorithms aim to solve for the near-optimal resource input strategy.

5.1 Centralized Algorithm CGBD

To address the case of full information sharing among clients, we propose the Centralized Generalized Benders Decomposition (CGBD) algorithm. As previously mentioned, solving for a potential game's Nash equilibrium involves finding the potential function's global maximums. Therefore, we formalize the problem as follows:

$$\begin{aligned} &\max_{\boldsymbol{\pi}} \quad \boldsymbol{U}(\boldsymbol{\pi}) \\ &\text{s.t. } C_i^{(1)}, C_i^{(2)}, C_i^{(3)}, \forall i \in \mathcal{N}, \end{aligned} \tag{8}$$

where $\boldsymbol{\pi} = \{\pi_i = \{d_i, f_i\}\}_{i\in\mathcal{N}}$ is the set of strategies for all clients. The problem (8) is a Mixed-integer Nonlinear Programming (MINLP) problem for any data-accuracy function $\boldsymbol{P}\left(d_i, \boldsymbol{d}_{-i}\right)$ mentioned in Section 3.2, where the presence of the f_i^2 term makes it non-convex. However, if f_i is fixed while $\boldsymbol{P}\left(d_i, \boldsymbol{d}_{-i}\right)$ is a convex function, then (8) can be transformed into a convex problem.

To solve (8) in a way that ensures near optimality, we propose the CGBD algorithm, which does not depend on any exact form of the data-precision function. Through CGBD, (8) can be decomposed into a *primal problem* and a *master problem*, as shown in (9) and (11), which can be solved alternatively. Denoting the number of iterations as k, the *primal Problem* can be expressed as follows:

$$\begin{aligned} &\min_{\boldsymbol{d}} -\boldsymbol{U}(\boldsymbol{d}, \boldsymbol{f}^{(k-1)}) \\ &\text{s.t. } d_i \in \mathcal{X}_i, \forall i \in \mathcal{N}, \\ &\quad T_i^{(1)} + T_i^{(2)}(d_i, f_i^{(k-1)}) + T_i^{(3)} \le \tau, \forall i \in \mathcal{N}, \end{aligned} \tag{9}$$

where $\mathcal{X}_i \triangleq [D_{min}, 1]$. According to Lemma 1, (9) is a convex problem. By solving (9), we can obtain the optimal value $-\boldsymbol{U}(\boldsymbol{d}^{(k)}, \boldsymbol{f}^{(k-1)})$ and Lagrange multiplier vectors $\boldsymbol{u}^{(k)}$. Therefore, the Lagrangian function of (9) can be represented as $\boldsymbol{\mathcal{L}}^*\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}, \boldsymbol{u}^{(k)}\right) = \boldsymbol{U}\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}\right) + \boldsymbol{u}^{(k)^T}\boldsymbol{G}\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}\right)$, where $\boldsymbol{G}\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}\right) = \{T_i^{(1)} + T_i^{(2)}(d_i^{(k)}, f_i) + T_i^{(3)} - \tau\}_{i\in\mathcal{N}}$. It is important to note that not all $\boldsymbol{f}^{(k-1)}$ lead to a feasible primal problem. If problem (9) is not feasible at particular values of $\boldsymbol{f}^{(k-1)}$, we consider the following problem:

$$\begin{aligned} &\min_{\{\boldsymbol{d},\zeta\}} \quad \zeta \\ &\text{s.t. } \boldsymbol{G}\left(\boldsymbol{d}, \boldsymbol{f}^{(k-1)}\right) \le \zeta, \boldsymbol{d} \in \boldsymbol{\mathcal{X}}, \zeta \ge 0, \end{aligned} \tag{10}$$

where $\boldsymbol{\mathcal{X}} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_{|\mathcal{N}|}, d_i \in \mathcal{X}_i$. Specifically, $\zeta > 0$ indicates that problem (9) is infeasible, otherwise it is feasible. By solving (10), we can also obtain its Lagrange multiplier vector $\boldsymbol{\lambda}^{(k)}$, which yields a Lagrangian function for (10), i.e., $\boldsymbol{\mathcal{L}}_*\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}, \boldsymbol{\lambda}^{(k)}\right) = \boldsymbol{\lambda}^{(k)^T}\left(\boldsymbol{G}\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}\right) - \zeta\right)$.

Let $\mathcal{V}_{fea}^{(k)}$ and $\mathcal{I}_{inf}^{(k)}$ denote the set of feasible and infeasible iteration indices, respectively. Then, the *master Problem* can be represented as:

$$\begin{aligned} &\min_{\{\boldsymbol{f},\varphi\}} \quad \varphi \\ &\text{s.t. } \varphi \ge \boldsymbol{\mathcal{L}}^*\left(\boldsymbol{d}_v^{(k)}, \boldsymbol{f}, \boldsymbol{u}_v^{(k)}\right), \forall v \in \mathcal{V}_{fea}^{(k)}, \boldsymbol{u}^{(k)} \succeq \boldsymbol{0}, \\ &\quad 0 \ge \boldsymbol{\mathcal{L}}_*\left(\boldsymbol{d}_v^{(k)}, \boldsymbol{f}, \boldsymbol{\lambda}_v^{(k)}\right), \forall v \in \mathcal{I}_{inf}^{(k)}, \boldsymbol{f} \in \boldsymbol{\mathcal{F}}, \end{aligned} \tag{11}$$

where $\boldsymbol{\mathcal{F}} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_{|\mathcal{N}|}, f_i \in \mathcal{F}_i \triangleq [F_i^{(1)}, \ldots, F_i^{(m)}]$. $\mathcal{V}_{fea}^{(k)}$ and $\mathcal{I}_{inf}^{(k)}$ are dynamically updated in each iteration. By alternatively solving the primal problem (9) and the master problem (11) until the upper bound $UB^{(k)}$ and the lower bound $LB^{(k)}$ satisfy the convergence condition, i.e. $UB^{(k)} - LB^{(k)} \le \epsilon$, where ϵ is the precision error, we can obtain the global solution to problem (8). The complete running process of CGBD is presented in Algorithm 1. Lemma 1 analyzes the convexity of (9) (proof omitted for simplicity).

Lemma 1 (Convexity Analysis)**.** *For any $\boldsymbol{f}^{(k-1)} \in \boldsymbol{\mathcal{F}}$, the primal problem defined in (9) is a convex problem.*

As per Lemma 1 (proof can be found in [1]), $\boldsymbol{d}^{(k)}$ and $\boldsymbol{u}^{(k)}$ can be acquired, and the corresponding value of the objective function $-\boldsymbol{U}(\boldsymbol{d}^{(k)}, \boldsymbol{f}^{(k-1)})$ can have a precision error within $\delta > 0$ [40]. Since problems (9) and (10) possess similar structures, the interior point method (IP) [40] can be also employed. The master problem defined in (11) is a mixed-integer programming (MIP) problem, which is tackled by enumerating the feasible values of $\boldsymbol{f}^{(k)}$ in this work. Here we directly present several lemmas regarding the convergence, optimality, and computational complexity of CGBD [1].

Lemma 2 (Termination within a finite number of iterations)**.** *The CGBD algorithm terminates in a finite number of steps for any given $\epsilon > 0$, even if $\epsilon = 0$.*

Lemma 3 (Near-optimality)**.** *The CGBD algorithm yields a solution $(\delta + \epsilon)$-optimal for the optimization problem (8).*

Lemma 4 (Complexity Analysis)**.** *When solving the master problem using the enumeration method, the computational complexity of Algorithm 1 is $\mathcal{O}(Im^{|\mathcal{N}|})$, where $I \le K$. When solving the master problem using the IP method, the computational complexity of Algorithm 1 is $O\left(\sqrt{|\mathcal{N}|m}\log\left(\frac{|\mathcal{N}|m}{\delta}\right)\right)$.*

The proofs of Lemma 1 to Lemma 4 can be found in [1]. It is worth noting that CGBD can theoretically achieve a near-optimal solution for the resource allocation strategy. *However, CGBD requires complete sharing of resource investment strategies $\pi_i, i \in \mathcal{N}$, profit factor $p_i, i \in \mathcal{N}$, energy consumption factor $\eta_i, i \in \mathcal{N}$, and other information among all clients to perform centralized optimization. Therefore, CGBD is applicable to scenarios with full global information sharing.*

5.2 Distributed Algorithm DBR

As demonstrated in Lemma 3, CGBD can result in $(\delta + \epsilon)$-optimal solutions. However, its execution requires centralized controllers and depends on global information sharing, which can be problematic for distributed clients. To address this issue and enable client autonomous decision-making in co-training scenarios with limited information sharing, we propose a distributed algorithm called *dynamic best response*

Algorithm 1: Centralized algorithm based on generalized Benders decomposition (CGBD)

Input: Iteration index $k = 0$, precision error ϵ, $UB^{(0)} = \infty, LB^{(0)} = -\infty$, and $\boldsymbol{f}^{(0)} \in \boldsymbol{\mathcal{F}}$.
Output: Near-global solution $\boldsymbol{\pi}^{NE} = \{\boldsymbol{d}^*, \boldsymbol{f}^*\}$.

while $UB^{(k)} - LB^{(k)} > \epsilon$ *and* $k < K$ **do**
  $k \leftarrow k + 1$;
  if *primal problem (9) is feasible* **then**
    Solve (9) with $\boldsymbol{f}^{(k-1)}$ to obtain $\boldsymbol{d}_v^{(k)}$, $\boldsymbol{u}_v^{(k)}$, $\boldsymbol{\mathcal{L}}^*(\boldsymbol{d}_v^{(k)}, \boldsymbol{f}, \boldsymbol{u}_v^{(k)})$;
  else
    Solve problem (10) to obtain $\boldsymbol{d}_v^{(k)}$, $\boldsymbol{u}_v^{(k)}$, and $\boldsymbol{\mathcal{L}}_*\left(\boldsymbol{d}_v^{(k)}, \boldsymbol{f}, \boldsymbol{\lambda}_v^{(k)}\right)$;
  $UB^{(k)} \leftarrow \min\{UB^{(k-1)}, \boldsymbol{U}(\boldsymbol{d}^{(k)}, \boldsymbol{f}^{(k-1)})\}$;
  Solve the master problem (11) to obtain $\boldsymbol{f}^{(k)}$, φ^*;
  Update the lower bound $LB^{(k)} \leftarrow \varphi^*$.
Return the global solution $\boldsymbol{\pi}^{NE} = \{\boldsymbol{d}^*, \boldsymbol{f}^*\}$.

(DBR) for weighted potential games, as introduced in [41]. The definition of *best response* is provided below.

Definition 8 (Dynamic best response for weighted potential games)**.** *Given other clients' strategies, $\boldsymbol{\pi}_{-i}$, o_i determines its best response (BR) as the resource investment strategy that maximizes its gains in this stage, which is expressed as*

$$\pi_i' = \underset{d_i \in \mathcal{X}_i, f_i \in \mathcal{F}_i}{\arg\max} \quad \boldsymbol{\mathcal{C}}_i\left(\pi_i, \boldsymbol{\pi}_{-i}\right). \tag{12}$$

The GBD-based algorithm can be utilized to obtain the best response of o_i by solving (12), which has a similar structure to (8). Then, by iteratively updating the players' strategies to the best responses, the Nash equilibrium of the potential game can be realized in finite steps [17]. The procedure of DBR is presented in Algorithm 2.

Notably, the difference between DBR and CGBD lies in the fact that DBR enables clients to autonomously determine their resource input $\boldsymbol{\pi}^{NE}$ without relying on a centralized parameter server, requiring only the resource input decision information of the clients. This is because the best-response strategy solution for each client does not depend on the cost information of other clients. As shown in Algorithm 2, solving the best-response strategy for DBR only requires sharing the decision information of the clients. In other words, DBR is suitable for scenarios where only partial information is shared. Additionally, even if $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ is non-convex, DBR can still obtain a resource input strategy under Nash equilibrium as long as (6) still satisfies the properties of the potential game.

As the convergence of BR-based algorithms has already been established [17], we focus solely on the computational complexity of Algorithm 2. Compared to CGBD, the complexity of DBR is reduced from exponential to polynomial, specifically, $\mathcal{O}(TL|\mathcal{N}|m)$, where, $T \leq H$ and $L \leq I$.

Theorem 2 (Individual rationality, budget balance, and computational efficiency.)**.** *The proposed incentive mechanism satisfies the conditions of individual rationality (IR), budget balance (BB), and computational efficiency (CE).*

Proof. Refer to **Appendix** for the detailed proof. □

5.3 Multi-Agent Stochastic Learning Algorithm

As discussed in Section 5.2, the DBR algorithm can achieve NE in polynomial time in a distributed manner, relying solely on the sharing of decision-making information among clients. However, in real-world CEC training scenarios, clients may still hesitate to share it. Moreover, if the data distribution among clients is non-uniform, it may be difficult to identify an explicit data-accuracy function $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ that accurately captures its correlation with $\{d_i, \boldsymbol{d}_{-i}\}$. Furthermore, network fluctuations and changes in computational and data resources owned by clients may cause dynamic variations in the training overhead. All these challenges necessitate adaptive incentive algorithms suitable for real co-training in information-unshared and dynamic environments.

Algorithm 2: Distributed algorithm based on dynamic best response (DBR)

Input: $\{\pi_i^{(0)}\}_{i \in \mathcal{N}}$, where $d_i^{(0)} = D_{min}, f_i^{(0)} = F_i^{(m)}$, decision time slot $t \leftarrow 0$.
Output: Resource input strategy profile of Nash equilibrium $\boldsymbol{\pi}^{NE}$.

while $t < H$ **do**
  $t \leftarrow t + 1$;
  Calculate each client o_i's best response π_i' with GBD algorithm;
  if $\pi_i' \neq \pi_i^{[t-1]}$ **then**
    Update o_i's strategy $\pi_i^{[t]} \leftarrow \pi_i'$;
  if *no client changes resource input strategy* **then**
    break;
return $\boldsymbol{\pi}^{NE} = \{\boldsymbol{d}^*, \boldsymbol{f}^*\}$.

Fortunately, distributed stochastic learning, the algorithm that enables game players to achieve the NE in a dynamic environment [42], provides a promising solution to the aforementioned challenges. *Therefore, we propose a Multi-Agent Stochastic Learning (MASL) algorithm, which can obtain approximately optimal resource contribution strategies in a distributed manner within dynamic environments. This is achieved without prior knowledge of the data-accuracy function and without the need to share resource contribution strategies with other clients, making it suitable for scenarios where information is not shared.*

In Algorithm 3, the resource input strategy for each client $\{\hat{\pi}_n^t\}_{t \in \mathcal{T}}$ is discrete. Specifically, $\hat{\pi}_n^t \triangleq \{\hat{d}_n^t, f_n^t\}$, where $\hat{d}_n^t \in \{D_{min}, D_{min} + e, \ldots, D_{min} + qe\}$, with $e \in \mathbb{R}$ as the step interval, $D_{min} + qe < 1$, $q \in \mathbb{N}$, and $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ being implicit. At each time slot $t \in \mathcal{T}$, the client o_i determines its resource inputs based on its strategy probability vector $\boldsymbol{w}_n^t \triangleq \{w_{n,\hat{\pi}_n}^t\}_{\hat{\pi}_n \in \mathcal{A}_n}$, where $\mathcal{A}_n$ is the space of o_n's strategy. Subsequently, o_n updates its probability vector to $\boldsymbol{w}_n^{t+1}$ with the corresponding reward, $\mathcal{C}_n^t(\hat{\pi}_n^t)$, obtained from the dynamic training environment at the $(t+1)$th time slot. Here, $lr_n^t = b\mathcal{H}_n\left(\mathcal{C}_n^t(\hat{\pi}_n^t)\right)$ represents an adaptive step size that varies with the time slot, where $b \in [0, 1]$ is the learning rate, and $\mathcal{H}_n(\cdot)$ is a function that maps the reward $\mathcal{C}_n^t(\hat{\pi}_n^t)$ to interval $[-1, 1]$, with $\boldsymbol{\varsigma}_{\hat{\pi}_n^t}$ as a unit vector whose $\hat{\pi}_n^t$th element is 1. The sampling probability of the optimal resource input policy, $\hat{\pi}_n^t$, for each o_n is increased if it yields higher rewards. Following multiple rounds of client-environment interactions, this probability of $\hat{\pi}_n^{NE}$ is updated to one.

Theorem 3 (Convergence rate)**.** *The number of iterations required for MASL to reduce $\mathcal{U}^{\boldsymbol{W}^0} - \mathcal{U}^{\boldsymbol{W}^{NE}}$ to $\frac{1}{|\mathcal{T}|}$ of its original value is $\frac{\log|\mathcal{T}|}{\log\bar{\rho}}$. Here, $\bar{\rho}$ represents the average convergence rate, $\boldsymbol{W} \triangleq \{\boldsymbol{w}_n\}_{n \in \mathcal{N}}$, and $\boldsymbol{W}^{NE} \triangleq \{\boldsymbol{w}_n^{NE}\}_{n \in \mathcal{N}}$ denotes the*

Algorithm 3: Multi-agent stochastic learning

Input: Time slot $t \leftarrow 0$, each client o_n's strategy probability vector $\boldsymbol{w}_n^t = (\frac{1}{V}, \ldots, \frac{1}{V})$, where $V = (q+1)m$.
Output: Near-optimal strategy probability vector $\boldsymbol{W}^{NE} \triangleq \{\boldsymbol{w}_n^{NE}\}_{n\in\mathcal{N}}$.

while $t = 1, 2, \ldots, |T|$ *and there are still client adjusting their resource input strategies* **do**
  Each client o_n decides the resource input strategy $\hat{\pi}_n^t = \{\hat{d}_n^t, f_n^t\}$ according $\boldsymbol{w}_n^t$.
  Each client o_n evaluates its respective reward $\mathcal{C}_n^t$ according to (5).
  Each client o_n updates the strategy probability vector $\boldsymbol{w}_n^t$: $\boldsymbol{w}_n^{t+1} = \boldsymbol{w}_n^t + lr^t(\varsigma_{\hat{\pi}_n^t} - \boldsymbol{w}_n^t)$.
Return $\boldsymbol{W}^{NE} \triangleq \{\boldsymbol{w}_n^{NE}\}_{n\in\mathcal{N}}$.

strategy probability vector under NE. The expectation of the potential function is defined as follows:

$$\mathcal{U}^{\boldsymbol{W}} = \mathbb{E}_{\hat{\boldsymbol{\pi}}\sim\boldsymbol{W},\theta\sim\eta}[\boldsymbol{U}(\hat{\boldsymbol{\pi}},\theta)]) = \sum_{\hat{\boldsymbol{\pi}}\in\mathcal{A}} \mathbb{E}_{\theta\sim\eta}[\boldsymbol{U}(\hat{\boldsymbol{\pi}},\theta)] \prod_{n\in\mathcal{N}} w_{n,\hat{\pi}_n}, \tag{13}$$

where $\hat{\boldsymbol{\pi}} \triangleq \{\hat{\pi}_n\}_{n\in\mathcal{N}}$, $\mathcal{A} \triangleq \{\mathcal{A}_n\}_{n\in\mathcal{N}}$, θ represents the environmental parameter, and η signifies its distribution.

Proof. Refer to **Appendix** for the detailed proof. □

5.4 Multi-Agent Decentralized Algorithm Based on Policy Gradient MAD-PG

Although MASL is suitable for scenarios without information sharing, its reliance on stochastic updates may lead to slow convergence in rapidly changing environments. Fortunately, techniques like importance sampling from off-policy enable an agent to better leverage the experience in episodic tasks [4], [43]. To ensure fast decision-making and reduce the number of iterations in highly dynamic settings, we propose a multi-agent decentralized algorithm based on policy gradient (MAD-PG). As detailed in Algorithm 4, MAD-PG is based on the policy gradient method, combining value function evaluation and policy function updates to achieve rapid and robust convergence in non-stationary environments. Next, we introduce the concept of average reward potential games, key properties of MAD-PG.

Definition 9 (Average reward potential games)**.** *An average reward potential game is a subset of an average reward Markov potential game (AMPG), in which there is a potential function $\Phi(\cdot)$ that holds the property that*

$$\Phi(W_i, \boldsymbol{W}_{-i}) - \Phi(W_i', \boldsymbol{W}_{-i}) = \rho_i^{W_i,\boldsymbol{W}_{-i}} - \rho_i^{W_i',\boldsymbol{W}_{-i}}, \tag{14}$$

where $\rho_i^{\boldsymbol{W}} \triangleq \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}_{\hat{\boldsymbol{\pi}}\sim\boldsymbol{W},\theta\sim\eta}[\sum_{t=0}^{T-1} \boldsymbol{\mathcal{C}}_i(\hat{\pi}_i, \hat{\boldsymbol{\pi}}_{-i}, \theta)]$ is the long-term average reward[9]*.*

To quantify the process of approaching the NE of the game, we define the Nash-gap and Nash-regret as

$$\begin{aligned} Nash\text{-}Gap(t) &\triangleq \max_i \max_{W_i} (\mathcal{R}_i^{W_i,\boldsymbol{W}_{-i}^t} - \mathcal{R}_i^{W_i^t,\boldsymbol{W}_{-i}^t}), \\ Nash\text{-}Regret(T) &\triangleq \frac{1}{T}\sum_{t=0}^{T-1} Nash\text{-}Gap(t), \\ Nash\text{-}Regret^*(T) &\triangleq \frac{1}{T}\sum_{t=0}^{T-1} [Nash\text{-}Gap(t)]^2. \end{aligned} \tag{15}$$

It is apparent that, with $Nash\text{-}Regret(T) \leq \epsilon$, the ϵ-Nash equilibrium can be achieved by selecting $t^* = \arg\min_t Nash\text{-}Gap(t)$ and $\boldsymbol{W} = \boldsymbol{W}^{t^*}$.

9. In this work, $\rho_i^{\hat{\boldsymbol{W}}} = \mathbb{E}_{\hat{\boldsymbol{\pi}}\sim\boldsymbol{W},\theta\sim\eta}[\mathcal{C}_i(\hat{\boldsymbol{\pi}},\theta)]$. Moreover, the potential function Φ is equivalent to $\mathcal{U}$, as defined in Section 5.3. For consistency, we denote $\rho_i^{\hat{\boldsymbol{W}}}$ as $\mathcal{R}_i^{\hat{\boldsymbol{W}}}$.

Algorithm 4: Multi-agent decentralized algorithm based on policy gradient

Input: $\{\alpha_i^t\}_{i\in\mathcal{N}}, \{\beta_i^t\}_{i\in\mathcal{N}}$, o_i's action space $\mathcal{A}_i$, buffer size M, batch size N, the parameters for strategy function network (π_i), ω_i^0, and value function network (V_i), ξ_i^0, and the buffers B_i initialized, the decision step $t \leftarrow 0$.
Output: The strategy set $\boldsymbol{\pi}^{NE}$.

while $t < T$ **do**
  Each o_i sample projected an action $a_i^t \in \mathcal{A}_i$ with its probability $\pi_i(a_i^t, \omega_i^t)$.
  Each o_i observes its utility $\boldsymbol{\mathcal{C}}_i^t$.
  Each o_i replaces the oldest record in B_i with $(a_i^t, \pi_i(a_i^t, \omega_i^t), \boldsymbol{\mathcal{C}}_i^t)$.
  Each o_i randomly selects N records from B_i.
  Each o_i updates its strategy function by
  $\omega_i^{t+1} \leftarrow \omega_i^t + \beta_i^t \sum_{j=1}^{N} (\boldsymbol{\mathcal{C}}_i^{t_j} - V_i(\xi_i^t)) \frac{\nabla_{\omega_i}\pi_i(a_i^{t_j},\omega_i^t)}{\pi_i(a_i^{t_j},\omega_i^{t_j})}$.
  Each o_i updates its value function by
  $\xi_i^{t+1} \leftarrow \xi_i^t + \alpha_i^t \sum_{j=1}^{N} (\boldsymbol{\mathcal{C}}_i^{t_j} - V_i(\xi_i^t)) \frac{\pi_i(a_i^{t_j},\omega_i^t)}{\pi_i(a_i^{t_j},\omega_i^{t_j})} \nabla_{\xi_i} V(\xi_i^t))$.
  $t \leftarrow t+1$.
Return $\boldsymbol{\pi}^{NE} \triangleq \{\arg\max_{a_i} \pi_i(a_i, w_i^t)\}_{i\in\mathcal{N}}$;

Lemma 5. *Suppose a function f is bounded below and L-smooth, i.e., $||\nabla f_\theta(\omega) - \nabla f_\theta(\omega')||_2 \leq L||\omega - \omega'||_2$, where θ is a random variable following a distribution given by η. Then, if we use stochastic gradient descent with a constant step size α small enough, i.e., $\omega^{k+1} = \omega^k - \alpha\nabla_\omega f_{\theta_k}|_{\omega^k}$, the error at iteration t is $\mathcal{O}(\frac{1}{\sqrt{t\alpha}})$.*

Proof. Please refer to **Appendix**. □

Theorem 4 (Bounded Nash-Regret analysis for MAD-PG's convergence)**.** *Suppose the environment has an $L_{\mathcal{U}}$-smooth potential function $\mathcal{U}(\cdot)$ and bounded variance for the reward functions. Let the learning rate of the policy function network be $\beta = \frac{1}{L_{\mathcal{U}}}$, and the learning rate of the value function network be a constant α. Define the lower bound of the strategy probability as p_{min}. When running Algorithm 4, the $Nash\text{-}Regret$ is bounded by $\mathcal{O}\left(\frac{1}{p_{min}\sqrt{T\alpha}}\right)$.*

Proof. Please refer to **Appendix**. □

In Algorithm 4, the strategy function networks $\{\pi_i\}_{i\in\mathcal{N}}$ have parameter sets $\{\omega_i^t\}_{i\in\mathcal{N}}$ to model the probability distributions of clients' resource input strategies, which can be in continuous or discrete spaces. Each client o_i has a value function network $V_i(\xi_i^t)$ to estimate the expected reward under the current strategy. At each time step $t \in \mathcal{T}$, client o_i samples an action a_i^t (corresponding to a resource input strategy) according to $\pi_i(a_i^t, \omega_i^t)$, observes the immediate reward $\mathcal{C}_i^t$, and stores the tuple $(a_i^t, \pi_i(a_i^t, \omega_i^t), \mathcal{C}_i^t)$ into its local buffer B_i. Subsequently, N records are randomly selected from B_i to update the strategy and value function networks using importance sampling and actor-critic methods. As the probability of selecting high-reward strategies increases with iterations, the approximate optimal strategy should correspond to the point with the highest long-term average reward. When the strategy distributions of each client gradually stabilize, their corresponding long-term average rewards and the potential function extrema ensure that the strategies remain at or near the approximate Nash Equilibrium.

6 Experiments and Results

In this section, we conduct experiments and implement a platform using Aliyun ECS.

TABLE I EXPERIMENT SETTINGS

Parameters	Values
p_i	$N(1000, 200)$
$\lvert\mathcal{S}_i\rvert$	$N(4 \times 10^3, 0.8 \times 10^3)$
f_i	$N(3 \times 10^9, 0.6 \times 10^9)$ Hz
κ	$[2 \times 10^{-28}, 5 \times 10^{-27}]$
CPU cycles per sample	5×10^6
Time constraint τ	$N(500, 100)$

6.1 Experimental Settings and Baselines

We conduct simulation experiments using a Dell server (Intel(R) Xeon(R) CPU E5-2630 v4@2.20GHz and 128GB RAM, with Ubuntu 18.04.6 as the operating system) to evaluate the performance of the proposed mechanism in terms of social welfare and convergence rate. To avoid restricting the proposed mechanism to a specific model-dataset combination, we utilize the accuracy loss function [15], [37] that has been widely employed in previous literature. We consider the following state-of-the-art baselines[10]:

- DBR Without Payoff Redistribution (WPR) [15]: The clients receive revenue solely from the performance of the global model, and the incentive does not involve the payoff redistribution term.
- DBR with Greedy Computation Allocation, (GCA) [28]: The clients greedily invest computational resources based on a linear multiple of the amount of data resources invested to meet the time constraint.
- Finite Improvement Property based Scheme (FIP) [44]: In this scheme, the Nash equilibrium is achieved through a finite set of update strategies. Unlike DBR, the data resource input variables under the FIP scheme are highly discrete.
- Optimal Resource Scheme (ALL-In): Each client provides all local data and computational resources regardless of training time constraints. [11]
- Better Response with inertia dynamics (BRI) [45]: This approach is model-free, meaning it does not make assumptions about the utility function. However, an amount of repetitive sampling of resource input strategies is required.

6.2 Smart Contract-Based Incentive Platform on Aliyun

We design and implement a prototype to evaluate the proposed mechanism's performance regarding real-world training efficiency. Specifically, a small simulation platform consisting of ten client instances and one model aggregation node instance equipped with 2.5GHz and 32vCPU is constructed using Aliyun ECS, as depicted in Fig. 5. We use the Pytorch 1.13.1 framework and CUDA 11.3 to co-train using three real model-dataset combinations, including ResNet18-MNIST, AlexNet-CIFAR10, and GoogleNet-SVHN. To ensure a fair comparison between different baseline schemes, a fixed total number of data samples is maintained for each model-dataset type in the local training stage. The computation resource allocation is primarily on GPUs by setting the application clock using the NVIDIA System Management Interface (Nvidia-semi-ac).

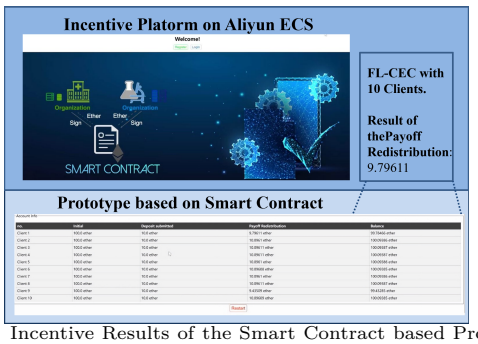


Fig. 5: Incentive Results of the Smart Contract based Prototype Implemented on Aliyun ECS.

Furthermore, we implement a customized incentive prototype on the private blockchain of Ethereum, based on the smart contract prototype designed in section 4.2. The prototype is written in 41 lines of Solidity and facilitates data interaction between the client and the smart contract through Web3 API, and real-time querying using web3.eth API.

6.3 Performance Analysis and Evaluation

Fig. 6 depicts the dynamic potential function value under various schemes. We observe that all schemes converge to the NE of the weighted potential game. CGBD achieves the highest potential function value with a relatively small gap with DBR. This result validates the proposed mechanism's effectiveness, which provides a fair payoff redistribution and develops a more efficient resource input strategy for each client.

Fig. 7 demonstrates the dynamic process of client payoff under DBR. In each iteration, DBR enables clients to autonomously determine their optimal amount of data and computational resource inputs. After approximately 25 iterations, the clients' returns gradually converge to a Nash equilibrium. Notably, DBR achieves the Nash equilibrium in a distributed manner and is suitable for co-training scenarios where information about resource input strategies is shared.

Fig. 8 displays the social welfare under different schemes. CGBD achieves the highest social welfare, followed by DBR. This is because CGBD and DBR steer the client's resource investment strategy towards the optimal direction by redistributing payoffs, thereby promoting higher social welfare compared to WPR. Furthermore, FIP can only provide a suboptimal resource investment strategy with a limited strategy space. Compared to GCA, DBR improves social welfare by approximately 18.4%. This is because, although GCA can satisfy the training time constraint, it determines the amount of computational resource inputs using a greedy approach, resulting in a suboptimal solution for the strategy.

Fig. 9 and Fig. 10 depict the effect of incentive intensity on social welfare under DBR and MAD-PG, respectively. As shown, when the incentive intensity increases from 4.78×10^{-10} to 1.13×10^{-8}, social welfare initially increases and then decreases. This demonstrates that a single increase in incentive intensity does not always improve social welfare. This is because while a higher incentive intensity may motivate the client to invest more resources in improving the global model accuracy, it also increases training overhead. Additionally, the optimal incentive intensity varies depending on the algorithm's characteristics. Moreover, even when evaluating

10. The comparisons between the proposed algorithms and the baselines are equitable as the total payoff redistribution in all schemes is zero and no supplementary incentives are introduced.

11. Since the value of the potential function and resource input strategy are fixed under this scheme, ALL-In lacks the dynamic interaction process of the potential game. Therefore, it is used as a comparison scheme for co-training performance, and only analyzed in the accuracy and training efficiency experiments.

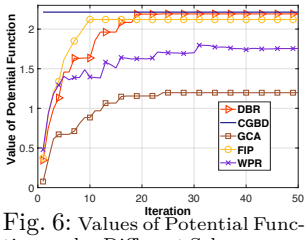


Fig. 6: Values of Potential Function under Different Schemes.

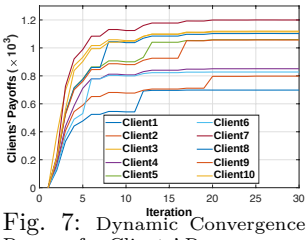


Fig. 7: Dynamic Convergence Process for Clients' Revenues.

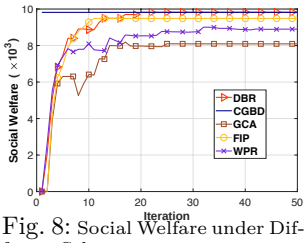


Fig. 8: Social Welfare under Different Schemes.

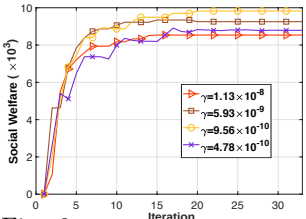


Fig. 9: Effect of Incentive Intensity on Social Welfare under DBR.

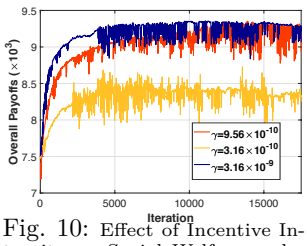


Fig. 10: Effect of Incentive Intensity on Social Welfare under MAD-PG.

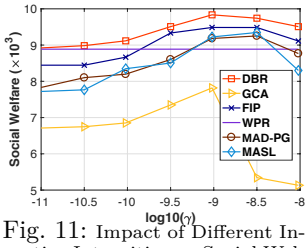


Fig. 11: Impact of Different Incentive Intensities on Social Welfare across Schemes.

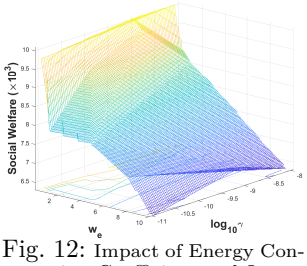


Fig. 12: Impact of Energy Consumption Coefficient and Incentive Intensity on Social Welfare under DBR.

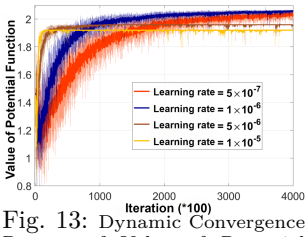


Fig. 13: Dynamic Convergence Process of Value of Potential Function at Different Learning Rates under MASL.

MAD-PG's performance based on the strategy profile with the highest probability, social welfare still experiences significant fluctuations during the near-convergence period due to MAD-PG's persistent exploration characteristic.

Fig. 11 demonstrates the impact of different incentive intensities on social welfare under other schemes. Since WPR does not redistribute payoffs, incentive intensity does not affect WPR. The figure shows that for all schemes except WPR, social welfare initially increases and then decreases with increasing incentive intensity, which supports the results in Fig. 9 and Fig. 10. Additionally, DBR achieves the highest social welfare through payoff redistribution and precise resource investment strategies compared to the baseline schemes. Furthermore, MASL slightly outperforms MAD-PG, and both achieve excellent social welfare close to FIP without sharing any information.

Fig. 12 depicts the joint effect of energy consumption coefficient and incentive intensity on social welfare under the DBR algorithm. The figure also shows a similar non-monotonic effect of the incentive intensity on social welfare. Additionally, as the energy consumption coefficient increases, social welfare decreases, demonstrating the non-negligible impact of training overhead on social welfare. By combining the results of Fig. 11 and Fig. 12, it can be observed that appropriate values of incentive intensity maximize social welfare.

Fig. 13 demonstrates the convergence process of the potential function under MASL, where each client adjusts its resource input strategy solely through reward feedback without sharing any strategy information among clients. Fig. 18 illustrates the dynamic process of the payoff of five clients when the number of clients N is 10 under MASL. It can be observed that, unlike Fig. 6, the convergence process of the potential function of MASL exhibits more drastic fluctuations caused by the rapid update of the resource input strategy in a dynamic environment. However, Fig. 13 indicates that MASL still achieves potential function values close to DBR despite the dynamic environment, demonstrating that MASL can effectively adapt to the dynamic environment and achieve adaptive incentives without shared decision information.

Fig. 14 illustrates the convergence of data resource investment strategy d_1 and computational resource investment strategy f_1 for client 1 during the operation of MASL. Interestingly, increased computational resource investment does not always enhance overall social welfare, as it also implies higher training costs. It can be observed from the figure that, despite the lack of information sharing among clients, they can still improve social welfare by adaptively updating their resource investment strategies during the MASL operation.

From a microscopic perspective, fig. 15-16 illustrate the evolution of strategies dynamics of clients under the proposed incentive mechanism. As the unit model performance gain coefficient of Client1 increases, other clients' resource input initially rises and stabilizes. This is because when p_1 increases, Client1 is motivated to invest more data to improve model accuracy, driven by the prospect of higher returns. Other clients benefit from the accuracy improvement. This indicates that the incentive mechanism effectively encourages resource investment, enhancing overall social welfare. Subsequently, due to diminishing marginal returns, the resource input of other clients tends to stabilize. The results in Fig. 17 clearly illustrate the impact of ϖ_e on the average data contribution $\bar{d}$. As ϖ_e increases, $\bar{d}$ consistently decreases across all schemes, reflecting that a higher ϖ_e discourages clients from investing data resources. Notably, WPR experiences a sharp decline in data contribution when $\varpi_e \geq 2.5$. In contrast, MASL and MAD-PG maintain higher $\bar{d}$ even under no information sharing, demonstrating the effectiveness of the proposed mechanism in incentivizing data contributions, even at higher ϖ_e values.

Fig. 19 examines the impact of the number of clients N on the average payoff. The average payoff is computed as the social welfare averaged over the number of clients. In this experiment, the average data size of the clients remains constant, resulting in a linear increase in total data input with the number of clients. As depicted in the figure, the average gain of clients increases as the number of clients increases from 4 to 15, but the growth rate slows down. This is because more data input enhances the accuracy performance of the global model, thereby increasing the client's gain from the global model. However, as data input continues to increase, there is a diminishing marginal utility of accuracy growth, and linear gain growth cannot be sustained, which is consistent with the

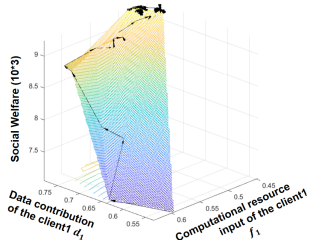


Fig. 14: Convergence Process of Client 1's Resource Input Strategy.

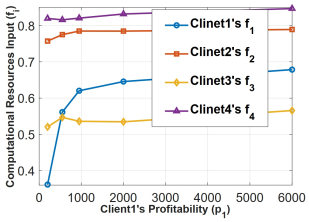


Fig. 15: Impact of p_i on the Computational Resources Input varies Across Clients.

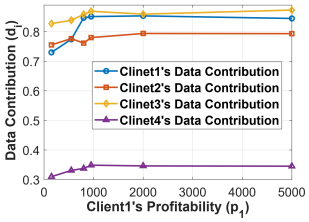


Fig. 16: Impact of p_i on the Data Contributions From Different Clients.

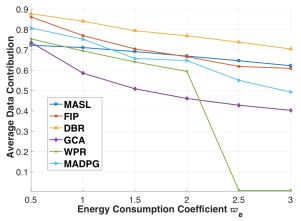


Fig. 17: Impact of ϖ_e on Average Data Contribution $\bar{d}$.

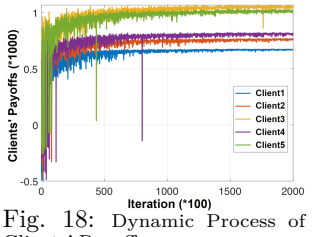


Fig. 18: Dynamic Process of Clients' Payoffs.

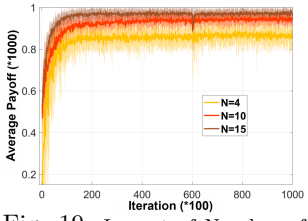


Fig. 19: Impact of Number of Clients on Average Revenue.

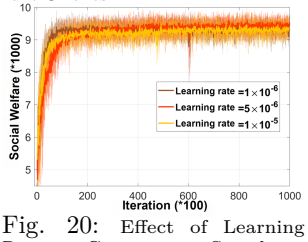


Fig. 20: Effect of Learning Rate on Convergence Speed under MASL.

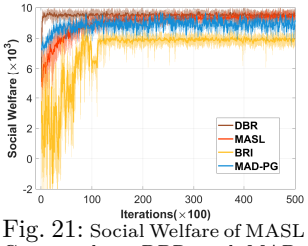


Fig. 21: Social Welfare of MASL Compared to DBR and MAD-PG.

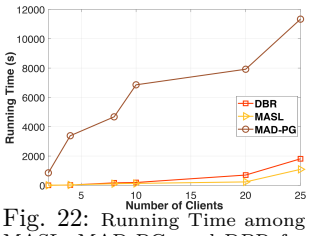


Fig. 22: Running Time among MASL, MAD-PG, and DBR for Different Number of Clients.

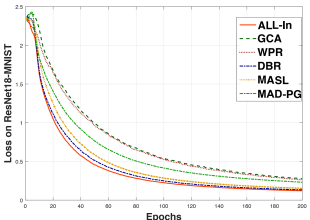


Fig. 23: Training Loss of Different Schemes on ResNet18-MNIST.

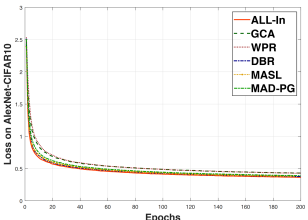


Fig. 24: Training Loss of Different Schemes on AlexNet-CIFAR10.

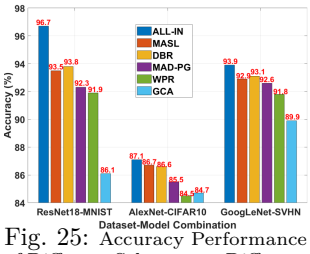


Fig. 25: Accuracy Performance of Different Schemes on Different Model-Datasets.

marginal effect of co-training.

Fig. 20 illustrates the impact of different learning rates on the convergence rate of MASL. The figure demonstrates that the effect of the learning rate on the MASL convergence rate is nonlinear, and a learning rate of 1×10^{-6} results in a relatively fast and stable convergence rate, which is consistent with the results of Fig. 13.

Fig. 21 displays the social welfare of different schemes in a dynamic co-training environment. The DBR algorithm is executed with information sharing among all clients, while MASL, MAD-PG, and BRI are without information sharing. In a dynamic collaborative environment, the same resource input strategy may result in different benefits due to the changing training overhead. The figure shows that even without shared information and a highly dynamic environment, MASL and MAD-PG achieve social welfare close to that of DBR. Moreover, although MASL requires more iterations to converge than MAD-PG, it achieves higher social welfare with a smaller variance at convergence. Additionally, compared to the BRI algorithm, which also does not require information sharing, MASL improves social welfare by about 15.6%. The results in Fig. 21 demonstrate that MASL can achieve excellent social welfare while realizing adaptive incentives even in a dynamic environment without shared information.

Fig. 22 displays the running time of the proposed MASL, MAD-PG, and DBR algorithms on the hardware platform and software environment described in Section 6.1. As the size of the clients grows from 2 to 25, the operation time of all algorithms increases accordingly. However, the running time of MASL is lower than that of DBR, and significantly lower than that of MAD-PG. This advantage becomes more pronounced as the number of clients increases. For instance, when there are 25 clients, MASL reduces the running time by up to 39.8% compared to DBR and even 90.4% compared to MAD-PG. The number of clients has less impact on the running time of the MASL algorithm compared to DBR since it does not require clients to exchange resource input strategies. Instead, clients obtain reward feedback directly from the environment and adaptively update the probability vector of the strategy to learn the near-optimal resource input strategy. In contrast, MAD-PG's deep-learning-based characteristics require high parallel computing performance and more iterations on updating parameters, even if it can sample less experience. As a result, when the frequency of interaction with the environment is high, MASL has a temporal advantage over MAD-PG.

Figs. 23 to 25 demonstrate the training efficiency and accuracy performance of the different schemes on real models and datasets. The results demonstrate that DBR improves both training efficiency and accuracy in comparison to FIP, WPR, and GCA. For instance, on ResNet18-MINIST, DBR enhances accuracy by 7.7% over WPR. Among all schemes, DBR's performance is closest to that of ALL-In. However, ALL-In overlooks training overhead, which restricts its application in resource-limited FL-CEC. Also, DBR compensates the client rationally through payoff redistribution, incentivizing them to invest more resources in achieving fast convergence and high accuracy. It is worth mentioning that on the AlexNet-CIFAR10 pair, MASL outperforms DBR in terms of both accuracy and training efficiency, albeit slightly. This is because, although MASL has a smaller search dimension than DBR in terms of the decision variable of data resource input, it is more

adaptive to fluctuations in training overhead in dynamic environments. Therefore, MASL may perform slightly better than DBR in such environments. This also explains the reason why MAD-PG's performance is below that of MASL, which is consistent with the results in Fig. 21. Moreover, despite the absence of clients' strategy information sharing, MASL achieves accuracy and training efficiency comparable to DBR on two model-dataset combinations, namely ResNet18-MINIST and GoogleNet-SVHN. In general, the DBR and MASL algorithms significantly enhance the accuracy performance of the trained global model compared to other baselines.

7 Conclusions

In this work, we design a novel incentive mechanism for FL-CEC and construct a multi-variable potential function based on a weighted potential game to describe the interaction of resource investment strategies. To handle the challenges of diversified information-sharing levels and environmental dynamics, we propose four algorithms: a GBD-based algorithm, a weighted potential game best-response algorithm, a distributed stochastic learning algorithm, and a policy gradient based algorithm, to solve for near-optimal resource input strategies. We analyze the convergence and complexity of the proposed algorithms. Furthermore, we develop an incentive platform using the Aliyun ECS, enabling automated incentive distribution through smart contracts. Extensive evaluation results demonstrate that the proposed scheme effectively enhances social welfare and improves the performance of collaborative training compared to baselines. In future work, we plan to integrate privacy-preserving protocols in FL to meet clients' data security and privacy needs.

8 Appendix

8.1 Proof of Theorem 1

Proof. Suppose o_i changes its strategy from $\pi_i = \{d_i, f_i\}$ to $\pi_i' = \{d_i', f_i'\}$, while the tuple of strategies for the other clients $\boldsymbol{\pi}_{-i}$ remains unchanged. In this case, the potential function takes the form $\boldsymbol{U}(\boldsymbol{\pi}')$, where

$$\begin{aligned}\boldsymbol{U}(\boldsymbol{\pi}') = \boldsymbol{P}\left(d_i', \boldsymbol{d}_{-i}\right) - \frac{\varpi_e \kappa f_i'^2 \eta_i d_i' |\mathcal{S}_i|}{p_i} + (n-1)\frac{\gamma r_i}{p_i} \\ - \sum_{j\in\mathcal{N}, j\neq i} \frac{\varpi_e \kappa f_j^2 \eta_i d_j |\mathcal{S}_j|}{p_i} + (n-1)\sum_{j\in\mathcal{N}, j\neq i}\frac{\gamma r_j}{p_j}.\end{aligned} \tag{16}$$

By subtracting (16) from (7), we obtain

$$\begin{aligned}&\boldsymbol{U}(\boldsymbol{\pi}) - \boldsymbol{U}(\boldsymbol{\pi}') \\ &= \frac{\boldsymbol{P}(d_i', \boldsymbol{d}_{-i})}{p_i} - \frac{\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})}{p_i} + \frac{\varpi_e \kappa f_i'^2 \eta_i d_i' |\mathcal{S}_i|}{p_i} - \frac{\varpi_e \kappa f_i^2 \eta_i d_i |\mathcal{S}_i|}{p_i} \\ &\quad + (n-1)\left(\frac{\gamma r'_i}{p_i} - \frac{\gamma r_i}{p_i}\right) \\ &= \frac{1}{p_i}\left[\boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}) - \boldsymbol{\mathcal{C}_i}(\pi_i', \boldsymbol{\pi}_{-i})\right],\end{aligned} \tag{17}$$

which completes the proof of Theorem 1. □

8.2 Proof of Theorem 2

Proof. Given $\widetilde{\pi}_i = \{D_{min}, f_i\}$ for $i \in \mathcal{N}$ and p_i, we can easily determine some γ so that $\mathcal{C}_i(\widetilde{\pi}_i, \boldsymbol{\pi}_{-i})$ remains non-negative. Under Nash equilibrium, it follows that $\mathcal{C}_i(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}) \geq \mathcal{C}_i(\widetilde{\pi}_i, \boldsymbol{\pi}_{-i}) \geq 0$, thereby satisfying individual rationality. As per (4), $\sum_{i\in\mathcal{N}} \boldsymbol{R}_i = 0$, indicating that the proposed mechanism satisfies budget balance without external incentives. Additionally, as stated earlier, the computational complexity of Algorithm 2 is $\mathcal{O}(TL|\mathcal{N}|m)$, ensuring a polynomial time execution for the incentives. Hence, the proposed mechanism is computationally efficient. □

8.3 Proof of Theorem 3

Proof. Given other clients' strategy probability vectors $\boldsymbol{W}_{-n}$, when o_n select strategy $\hat{\pi}_n$, the expectation of o_n's payoff can be computed by using (5) as follows:

$$\begin{aligned}\mathcal{R}_n^{\boldsymbol{W}_{-n}}(\hat{\pi}_n) &= \mathbb{E}_{\hat{\boldsymbol{\pi}}_{-n}\sim\boldsymbol{W}_{-n},\theta\sim\eta}[\boldsymbol{\mathcal{C}_n}(\hat{\pi}_n, \hat{\boldsymbol{\pi}}_{-n}, \theta)] \\ &= \sum_{\hat{\boldsymbol{\pi}}_{-n}\in\mathcal{A}_{-n}} \mathbb{E}_{\theta\sim\eta}[\boldsymbol{\mathcal{C}_n}(\hat{\pi}_n, \hat{\boldsymbol{\pi}}_{-n}, \theta)] \prod_{n'\neq n} w_{n',\hat{\pi}_{n'}},\end{aligned} \tag{18}$$

where $w_{n',\hat{\pi}_{n'}}$ is $o_{n'}$'s probability of selecting the strategy $\hat{\pi}_{n'}$. Correspondingly, the expected value of the potential function for it can be calculated as

$$\mathcal{U}_n^{\boldsymbol{W}_{-n}}(\hat{\pi}_n) = \mathbb{E}_{\hat{\boldsymbol{\pi}}_{-n}\sim\boldsymbol{W}_{-n},\theta\sim\eta}[\boldsymbol{U}(\hat{\pi}_n, \hat{\boldsymbol{\pi}}_{-n}, \theta)]. \tag{19}$$

Using (19) and the convergence theorem [42] and with $\mathcal{H}_n(\cdot) = \gamma_n(\cdot)$, we can calculate the difference in the expectation of the potential function between two-time slots as $\mathcal{U}^{\boldsymbol{W}^{t+1}} - \mathcal{U}^{\boldsymbol{W}^t} = b\sum_{n\in\mathcal{N}}(\gamma_n \mathrm{Cov}_{\hat{\pi}_n\sim W_n^t}[\mathcal{U}_n^{\boldsymbol{W}_{-n}^t}(\hat{\pi}_n), \mathcal{R}_n^{\boldsymbol{W}_{-n}^t}(\hat{\pi}_n)])$.

Given the weighted potential game [12], Theorem 1 implies

$$\mathcal{U}^{\boldsymbol{W}^{t+1}} - \mathcal{U}^{\boldsymbol{W}^t} = b\sum_{n\in\mathcal{N}}\gamma_n p_n \mathrm{Var}_{\hat{\pi}_n\sim W_n^t}[\mathcal{U}_n^{\boldsymbol{W}_{-n}^t}(\hat{\pi}_n)]. \tag{20}$$

With (20), the average convergence rate $\bar{\rho}$ can be computed as $\bar{\rho} = \sqrt{\rho^{t=0}\rho^{t\to\infty}}$, where ρ^t is defined as:

$$\begin{aligned}\rho^t &= \frac{\mathcal{U}^{\boldsymbol{W}^{t+1}} - \mathcal{U}^{\boldsymbol{W}^{NE}}}{\mathcal{U}^{\boldsymbol{W}^t} - \mathcal{U}^{\boldsymbol{W}^{NE}}} = 1 + \frac{\mathcal{U}^{\boldsymbol{W}^{t+1}} - \mathcal{U}^{\boldsymbol{W}^t}}{\mathcal{U}^{\boldsymbol{W}^t} - \mathcal{U}^{\boldsymbol{W}^{NE}}} \\ &= 1 + \frac{b\sum_{n\in\mathcal{N}}\gamma_n p_n(\mathbb{E}_{\hat{\pi}_n\sim W_n^t}[(\mathcal{U}_n^{\boldsymbol{W}_{-n}^t}(\hat{\pi}_n))^2] - (\mathcal{U}^{\boldsymbol{W}^t})^2}{\mathcal{U}^{\boldsymbol{W}^t} - \mathcal{U}^{\boldsymbol{W}^{NE}} + \mathcal{O}(b)}.\end{aligned} \tag{21}$$

Using $w_{n,\hat{\pi}_n}^0 = \frac{1}{(q+1)m}$, we can compute ρ^0 as:

$$\begin{aligned}\rho^0 = 1 + \frac{b}{2(q+1)^N m^N \sum_{\hat{\boldsymbol{\pi}}\in\mathcal{A}}(\boldsymbol{U}(\hat{\boldsymbol{\pi}}) - \mathcal{U}^{\boldsymbol{W}^{NE}})}\sum_{n\in\mathcal{N}}\gamma_n p_n \\ \sum_{\hat{\pi}_n, \hat{\pi}_n'\in\mathcal{A}_n}\Big(\sum_{\hat{\boldsymbol{\pi}}_{-n}\in\mathcal{A}_{-n}}(\boldsymbol{U}(\hat{\pi}_n, \hat{\boldsymbol{\pi}}_{-n}) - \boldsymbol{U}(\hat{\pi}_n', \hat{\boldsymbol{\pi}}_{-n}))\Big)^2.\end{aligned} \tag{22}$$

When $t \to \infty$, we can assume, for simplicity, that $\lim_{t\to\infty} w_{n,\hat{\pi}_n^{NE}}(t) = 1-\varepsilon$ and $\lim_{t\to\infty} w_{n,\hat{\pi}_n\neq\hat{\pi}_n^{NE}}(t) = \frac{\varepsilon}{(q+1)m-1}$, where $\varepsilon \to 0^+$. Then

$$\rho^\infty \approx 1 + b\frac{\sum_{n\in\mathcal{N}}\gamma_n p_n \sum_{\hat{\pi}_n\in\mathcal{A}_n}(\mathcal{U}_n^{\boldsymbol{W}_{-n}^{NE}}(\hat{\pi}_n) - \mathcal{U}^{\boldsymbol{W}^{NE}})^2}{\sum_{n\in\mathcal{N}}\sum_{\hat{\pi}_n\in\mathcal{A}_n}(\mathcal{U}_n^{\boldsymbol{W}_{-n}^{NE}}(\hat{\pi}_n) - \mathcal{U}^{\boldsymbol{W}^{NE}})}. \tag{23}$$

Consequently, we can determine the number of iterations required for MASL to reduce $\mathcal{U}^{\boldsymbol{W}^0} - \mathcal{U}^{\boldsymbol{W}^{NE}}$ to $\frac{1}{|\mathcal{T}|}$ times its original value as $\frac{\log|\mathcal{T}|}{\log\bar{\rho}}$, which completes the proof. □

8.4 Proof of Lemma 5

Proof. Since f is L-smooth, we have

$$\begin{aligned}\mathbb{E}_{\theta\sim\eta}\left[f_\theta(\omega^{k+1})\right] &\leq f_{\theta_k}(\omega^k) - \alpha||\nabla_\omega f_{\theta_k}|_{\omega^k}||_2^2 \\ &\quad + \alpha^2\frac{L}{2}\mathbb{E}_{\theta\sim\eta}\left[||\nabla_\omega f_\theta|_{\omega^k}||_2^2\right].\end{aligned} \tag{24}$$

Then there is

$$\begin{aligned}\sum_{k=1}^{t}(\alpha - \alpha^2\frac{L}{2})\mathbb{E}_{\theta\sim\eta}\left[||\nabla_\omega f_\theta|_{\omega^k}||^2\right] &\leq \sum_{k=1}^{t}\mathbb{E}_{\theta\sim\eta}\left[f_\theta(\omega^{k-1})\right] \\ - \sum_{k=1}^{t}\mathbb{E}_{\theta\sim\eta}\left[f_\theta(\omega^k)\right],\end{aligned} \tag{25}$$

which indicates

$$\min_{k=0,1,\ldots,t-1}\mathbb{E}_{\theta\sim\eta}\left[||\nabla_\omega f_\theta|_{\omega^k}||^2\right] \leq \frac{\mathbb{E}_{\theta\sim\eta}\left[f_\theta(\omega^0) - f_\theta(\omega^*)\right]}{k(\alpha - \alpha^2\frac{L}{2})}, \tag{26}$$

12. Our proposed MASL algorithm is also applicable to ordinal potential games, where $\boldsymbol{U}(\boldsymbol{\pi}) > \boldsymbol{U}(\boldsymbol{\pi}') \Leftrightarrow \boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}) > \boldsymbol{\mathcal{C}_i}(\pi_i', \boldsymbol{\pi}_{-i})$. This property indicates a positive covariance between them.

where ω^* is the optimal value that minimizes f.
With (26), an $\mathcal{O}(\frac{1}{\sqrt{t\alpha}})$-stationary point exists among the t iterations. □

8.5 Proof of Theorem 5.4

Proof. Denote $W_i^{*,t}$ as o_i's optimal strategy probability vector and $\widetilde{\boldsymbol{W}}^t$ as that updated with real gradient. For convenience, let client o_i have the largest Nash gap. Then according to the work [[46] Theorem 1], we have

$$Nash\text{-}regret^*_{unbiased}(T) = \frac{1}{T}\sum_{t=0}^{T-1}(\mathcal{R}_i^{W_i^{*,t},\widetilde{\boldsymbol{W}}_{-i}} - \mathcal{R}_i^{\widetilde{\boldsymbol{W}}})^2 \leq \frac{32L_{\mathcal{U}}C_{\mathcal{U}}}{T} \tag{27}$$

where $C_{\mathcal{U}} = \max_{\boldsymbol{W},\boldsymbol{W}'}|\mathcal{U}(\boldsymbol{W}) - \mathcal{U}(\boldsymbol{W}')|$. Hence,

$$\begin{aligned} Nash\text{-}regret(T) &= \frac{1}{T}\sum_{t=0}^{T-1}|\mathcal{R}_i^{W_i^{*,t},\boldsymbol{W}_{-i}^t} - \mathcal{R}_i^{\boldsymbol{W}^t}| \\ &\leq \frac{1}{T}\sum_{t=0}^{T-1}(|\mathcal{R}_i^{W_i^{*,t},\boldsymbol{W}_{-i}^t} - \mathcal{R}_i^{\tilde{W}_i^t,\boldsymbol{W}_{-i}^t}| + |\mathcal{R}_i^{\tilde{W}_i^t,\boldsymbol{W}_{-i}^t} - \mathcal{R}_i^{\boldsymbol{W}^t}|) \\ &\leq \sqrt{\frac{32L_{\mathcal{U}}C_{\mathcal{U}}}{T}} + \frac{1}{T}\sum_{t=0}^{T-1}|\mathcal{R}_i^{\widetilde{\boldsymbol{W}}^t} - \mathcal{R}_i^{\boldsymbol{W}^t}|, \end{aligned} \tag{28}$$

where $\boldsymbol{W}^t$ is the strategy learned from the sampled gradient.

$$\begin{aligned} \mathcal{R}_i^{\widetilde{\boldsymbol{W}}^{t+1}} - \mathcal{R}_i^{W_i^{t+1},\boldsymbol{W}_{-i}^{t+1}} &= \left\langle \overline{A_i}^{W^{t+1}}, \tilde{W}_i^{\,t+1} - W_i^{t+1} \right\rangle \\ &\leq \kappa||\tilde{W}_i^{\,t+1} - W_i^{t+1}||_1 \leq \kappa\beta||\tilde{g}_i^t - \hat{g}_i^t||_1, \end{aligned} \tag{29}$$

where the expectation of advantage function is defined as $\overline{A_i}^{\boldsymbol{W}}(\hat{\pi}_i) \triangleq \mathcal{R}_i^{\boldsymbol{W}_{-i}}(\hat{\pi}_i) - \mathcal{R}_i^{\hat{\boldsymbol{W}}}$, with $\kappa \triangleq \max_{i,\boldsymbol{W}}\min_{b\in\mathbb{R}}||\overline{A_i}^{\boldsymbol{W}} + b||_\infty$, which evaluates its deviation. If the strategy $\hat{\pi}_i$ is selected N times in a batch, then

$$\begin{aligned} ||\tilde{g}_i^t - \hat{g}_i^t||_1 &\leq \frac{A_{max}}{p_{min}}\max_{\hat{\pi}_i}|\overline{A_i}^{\boldsymbol{W}^t}(\hat{\pi}_i) - \overline{\boldsymbol{\mathcal{C}}}_{i,\hat{\pi}_i} + \hat{\mathcal{R}}_i^{\boldsymbol{W}^t}(\xi^t)| \\ &\leq \frac{A_{max}}{p_{min}}\max_{\hat{\pi}_i}(|\mathcal{R}_i^{\boldsymbol{W}_{-i}}(\hat{\pi}_i) - \overline{\boldsymbol{\mathcal{C}}}_{i,\hat{\pi}_i}| + |\mathcal{R}_i^{\boldsymbol{W}^t} - \hat{\mathcal{R}}_i^{\boldsymbol{W}^t}(\xi^t)|, \end{aligned} \tag{30}$$

where $\overline{\boldsymbol{\mathcal{C}}}_{i,\hat{\pi}_i} \triangleq \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{\mathcal{C}}_i(\hat{\pi}_i, \hat{\boldsymbol{\pi}}_{-i,k}, \theta_k)$ and and $\hat{\mathcal{R}}_i^{\boldsymbol{W}^t}(\xi^t)$ is the estimated long-term average reward at time slot t with parameter ξ^t. The upper bound of variance due to the dynamic environment is defined as $\sigma^2 \triangleq \max_i \max_{\hat{\pi}_i,\boldsymbol{W}_{-i}} \mathrm{Var}_{\hat{\boldsymbol{\pi}}_{-i}\sim\boldsymbol{W}_{-i},\theta\sim\eta}[\mathcal{C}_i(\hat{\pi}_i,\hat{\boldsymbol{\pi}}_{-i},\theta)]$, $\overline{\mathcal{C}}_{i,\hat{\pi}_i}$ almost follows a normal distribution with variance less than $\frac{\sigma^2}{N}$, with a large N according to the central limit theorem. Additionally, with Lemma 5 and taking $f_\theta(\xi^t) = (\mathcal{C}_i(\hat{\pi}_n^t, \hat{\boldsymbol{\pi}}_{-n}^t, \theta) - \hat{\mathcal{R}}_i^{\boldsymbol{W}^t}(\xi^t))^2$ with some L, in this case,

$$\mathbb{E}_{\theta\sim\eta}\left[||\tilde{g}_i^t - \hat{g}_i^t||_1\right] \leq \frac{A_{max}}{p_{min}}\left(\frac{\sigma}{\sqrt{N}} + \mathcal{O}(\frac{1}{\sqrt{t\alpha}})\right). \tag{31}$$

Otherwise, $||\tilde{g}_i^t - \hat{g}_i^t||_1 \leq \frac{A_{max}}{p_{min}}\max_{\hat{\pi}_i}|\overline{A_i}^{\boldsymbol{W}^t}(\hat{\pi}_i)| \leq \frac{\kappa A_{max}}{p_{min}}$. In conclusion,

$$\begin{aligned} \mathbb{E}_{\theta\sim\eta}[Nash\text{-}Regret(T)] &\leq \mathcal{O}(\sqrt{\frac{1}{T}}) + \mathcal{O}(\frac{1}{p_{min}}) \\ &+ \mathcal{O}(\frac{1}{p_{min}\sqrt{T\alpha}}) = \mathcal{O}(\frac{1}{p_{min}\sqrt{T\alpha}}). \end{aligned} \tag{32}$$

Although this proof is for the exact potential game with discrete action spaces, it also applies to weighted potential games or some ordinal potential games even with continuous action space similarly. □

References

[1] S. Yuan, H. Lv, H. Liu, C. Wu, S. Guo, Z. Liu, H. Chen, and J. Li, "Tradefl: A trading mechanism for cross-silo federated learning," in *Proc. IEEE ICDCS*, 2023, pp. 920–930.

[2] Y. Wei, S. Zhou, S. Leng, S. Maharjan, and Y. Zhang, "Federated learning empowered end-edge-cloud cooperation for 5g hetnet security," *IEEE Network*, vol. 35, no. 2, pp. 88–94, 2021.

[3] Amgen and Nvidia. (2022) The machine learning ledger orchestration for drug discovery (melloddy) project. [Online]. Available: https://www.melloddy.eu/

[4] S. Yuan, B. Dong, H. Lvy, H. Liu, H. Chen, C. Wu, S. Guo, Y. Ding, and J. Li, "Adaptive incentivize for cross-silo federated learning in iiot: A multi-agent reinforcement learning approach," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 15 048–15 058, 2023.

[5] X. Wang, Y. Zhao, C. Qiu, Z. Liu, J. Nie, and V. C. Leung, "Infedge: A blockchain-based incentive mechanism in hierarchical federated learning for end-edge-cloud communications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3325–3342, 2023.

[6] Y. Zhan, P. Li, S. Guo, and Z. Qu, "Incentive mechanism design for federated learning: Challenges and opportunities," *IEEE Network*, vol. 35, no. 4, pp. 310–317, 2021.

[7] H. Lv, Z. Zheng, T. Luo, F. Wu, S. Tang, L. Hua, R. Jia, and C. Lv, "Data-free evaluation of user contributions in federated learning," in *Proc. WiOpt*, 2021, pp. 81–88.

[8] Y. Xue, C. Niu, Z. Zheng, S. Tang, C. Lv, F. Wu, and G. Chen, "Toward understanding the influence of individual clients in federated learning," in *Proc. AAAI*, 2021, pp. 10 560–10 567.

[9] X. Wu and H. Yu, "Mars-fl: Enabling competitors to collaborate in federated learning," *IEEE Trans. Big Data*, vol. 14, no. 8, pp. 30–41, 2022.

[10] M. R. Behera, S. Upadhyay, and S. Shetty, "Federated learning using smart contracts on blockchains, based on reward driven approach," *arXiv preprint arXiv:2107.10243*, 2021.

[11] N. Ding, Z. Fang, L. Duan, and J. Huang, "Incentive mechanism design for federated learning with multi-dimensional private information," in *Proc. WiOpt*, 2020, pp. 33–40.

[12] P. Sun, H. Che, Z. Wang, Y. Wang, T. Wang, L. Wu, and H. Shao, "Pain-fl: Personalized privacy-preserving incentive for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3805–3820, 2021.

[13] J. Zhang, Y. Wu, and R. Pan, "Incentive mechanism for horizontal federated learning based on reputation and reverse auction," in *Proc. ACM WWW*, 2021, pp. 947–956.

[14] N. Zhang, Q. Ma, and X. Chen, "Enabling long-term cooperation in cross-silo federated learning: A repeated game perspective," *IEEE Trans. Mob. Comput.*, vol. 22, no. 7, pp. 3910–3924, 2023.

[15] M. Tang and V. W. Wong, "An incentive mechanism for cross-silo federated learning: A public goods perspective," in *Proc. IEEE INFOCOM*, 2021, pp. 1–10.

[16] Y. Li, X. Wang, R. Zeng, M. Yang, K. Li, M. Huang, and S. Dustdar, "Varf: An incentive mechanism of cross-silo federated learning in mec," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15 115–15 132, 2023.

[17] Y. H. Chew, B.-H. Soong *et al.*, *Potential Game Theory*. Springer, 2016.

[18] T. H. T. Le, N. H. Tran, Y. K. Tun, M. N. Nguyen, S. R. Pandey, and C. S. Hong, "An incentive mechanism for federated learning in wireless cellular networks: An auction approach," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 8, pp. 4874–4887, 2021.

[19] Y. Yuan, L. Jiao, K. Zhu, and L. Zhang, "Incentivizing federated learning under long-term energy constraint via online randomized auctions," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 7, pp. 5129–5144, 2021.

[20] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, P. Yang, Y. Zhou, and Y. Zhang, "Fair: Quality-aware federated learning with precise user incentive and model aggregation," in *Proc. IEEE INFOCOM*, 2021, pp. 1–10.

[21] J. S. Ng, W. Y. B. Lim, Z. Xiong, X. Cao, D. Niyato, C. Leung, and D. I. Kim, "A hierarchical incentive design toward motivating participation in coded federated learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 359–375, 2021.

[22] L. U. Khan, S. R. Pandey, N. H. Tran, W. Saad, Z. Han, M. N. Nguyen, and C. S. Hong, "Federated learning for edge networks: Resource optimization and incentive mechanism," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 88–93, 2020.

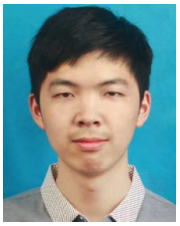
[23] N. Ding, Z. Fang, and J. Huang, "Optimal contract design for efficient federated learning with multi-dimensional private information," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 186–200, 2020.

[24] M. Chen, Y. Liu, W. Shen, Y. Shen, P. Tang, and Q. Yang, "A mechanism design approach for multi-party machine learning," *Theoretical Computer Science*, vol. 1003, p. 114618, 2024.

[25] S. Yuan, J. Li, H. Chen, Z. Han, C. Wu, and Y. Zhang, "Jira: Joint incentive design and resource allocation for edge-based real-time video streaming systems," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 5, pp. 2901–2916, May 2023.

[26] S. Yuan, J. Li, and C. Wu, "Jora: Blockchain-based efficient joint computing offloading and resource allocation for edge video streaming systems," *J. Syst. Archit.*, vol. 133, no. 40, pp. 1–12, 2022.
[27] S. Kong, Y. Li, and H. Zhou, "Incentivizing federated learning," 2022. [Online]. Available: https://arxiv.org/abs/2205.10951
[28] J. Chen and H. Jiang, "Social welfare maximization in cross-silo federated learning," in *Proc. IEEE ICASSP*, 2022, pp. 4258–4262.
[29] S. Yuan, J. Li, J. Liang, Y. Zhu, X. Yu, J. Chen, and C. Wu, "Sharding for blockchain based mobile edge computing system: A deep reinforcement learning approach," in *Proc. IEEE GLOBECOM*, 2021, pp. 1–6.
[30] S. Q. Zhang, J. Lin, and Q. Zhang, "A multi-agent reinforcement learning approach for efficient client selection in federated learning," in *Proc. AAAI*, 2022, pp. 9091–9099.
[31] J. Zhang, S. Guo, D. Zeng, Y. Zhan, and R. Akerkar, "Adaptive federated learning on non-iid data with resource constraint," *IEEE Trans. Comput.*, vol. 71, no. 7, pp. 1655–1667, 2021.
[32] W. Zhang, D. Yang, W. Wu, H. Peng, N. Zhang, H. Zhang, and X. Shen, "Optimizing federated learning in distributed industrial iot: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, 2021.
[33] Y. Zhan, P. Li, L. Wu, and S. Guo, "L4l: Experience-driven computational resource control in federated learning," *IEEE Trans. Comput.*, vol. 71, no. 4, pp. 971–983, 2021.
[34] J. Zheng, K. Li, N. Mhaisen, W. Ni, E. Tovar, and M. Guizani, "Exploring deep-reinforcement-learning-assisted federated learning for online resource allocation in privacy-preserving edgeiot," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21 099–21 110, 2022.
[35] S. Arisdakessian, O. A. Wahab, A. Mourad, and H. Otrok, "Coalitional federated learning: Improving communication and training on non-iid data with selfish clients," *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 2462–2476, 2023.
[36] A. Agarwal and Dahleh, "A marketplace for data: An algorithmic solution," in *Proc. ACM EC*, 2019, pp. 701–726.
[37] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, and Stich, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. ACM ICML*, 2020, pp. 5132–5143.
[38] P. Helber, B. Bischke, and Dengel, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, vol. 12, no. 7, pp. 2217–2226, 2019.
[39] X. Yao, X. Wang, S.-H. Wang, and Y.-D. Zhang, "A comprehensive survey on convolutional neural network in medical image analysis," *Multimed. Tools. Appl.*, vol. 81, no. 29, pp. 41 361–41 405, 2022.
[40] F. A. Potra and S. J. Wright, "Interior-point methods," *J. Comput. Appl. Math.*, vol. 124, no. 1-2, pp. 281–302, 2000.
[41] S. Yuan, Y. Liu, S. Guo, J. Li, H. Chen, C. Wu, and Y. Yang, "Efficient online computing offloading for budget-constrained cloud-edge collaborative video streaming systems," *IEEE Transactions on Cloud Computing*, vol. 13, no. 1, pp. 273–287, Jan. 2025.
[42] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. Mob. Comput.*, vol. 18, no. 4, pp. 771–786, 2018.
[43] S. Yuan, Q. Zhou, J. Li, S. Guo, H. Chen, C. Wu, and Y. Yang, "Adaptive incentive and resource allocation for blockchain-supported edge video streaming systems: A cooperative learning approach," *IEEE Trans Mob Comput.*, vol. 24, no. 2, pp. 539–556, Feb. 2025.
[44] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.
[45] T. Q. Dinh, Q. D. La, T. Q. Quek, and H. Shin, "Learning for computation offloading in mobile edge computing," *IEEE Trans. Comput.*, vol. 66, no. 12, pp. 6353–6367, 2018.
[46] M. Cheng, R. Zhou, P. Kumar, and C. Tian, "Provable policy gradient methods for average-reward markov potential games," in *Proc. PMLR AISTATS*, 2024, pp. 4699–4707.

Shijing Yuan (Member, IEEE) received his Ph.D. from Shanghai Jiao Tong University and was honored as Outstanding Doctoral Graduate, Shanghai, China, in 2024. From 2023 to 2024, he was a Visiting Ph.D. student at Hong Kong Polytechnic University. He is with China Telecom Research Institute, Shanghai, China. His research interests include on-orbit computing for large-scale LEO networks, edge AI, and reinforcement learning.



Beiyu Dong is currently pursuing the Bachelor's degree in Electrical and Computer Engineering in University of Michigan-Shanghai Jiao Tong University Joint Institute (UM-SJTU JI), Shanghai, China. His current research interests include reinforcement learning, and game theory.

Jie Li (Fellow, IEEE) received the B.E. degree in computer science from Zhejiang University, Hangzhou, China, the M.E. degree in electronic engineering and communication systems from China Academy of Posts and Telecommunications, Beijing, China. He received the Dr. Eng. degree from the University of Electro-Communications, Tokyo, Japan. He is with Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China where he is a professor. His current research interests are in big data, IoT, blockchain, edge computing, OS, modeling and performance evaluation of information systems. He was a full professor in Department of Computer Science, University of Tsukuba, Japan. He was a visiting Professor in Yale University, USA, Inria Sophia Antipolis and Inria Grenoble-Rhone-Aples, France. He is the co-chair of IEEE Technical Community on Big Data and the founding Chair of IEEE ComSoc Technical Committee on Big Data and the Co-Chair of IEEE Big Data Community. He serves as an associated editor for many IEEE journals and transactions. He has also served on the program committees for several international conferences.

Song Guo (Fellow, IEEE) is currently a full professor in the Computer Science and Engineering at the Hong Kong University of Science and Technology. He is a fellow of the Canadian Academy of Engineering (FCAE) and an ACM Distinguished member. He is the editor-in-chief of the IEEE Open Journal of the Computer Society, and chair of the IEEE Communications Society (ComSoc) Space and the Satellite Communications Technical Committee (SSCTC). He was recognized as a highly cited researcher (Clarivate Web of Science). He was the recipient of more than 12 recognition and best paper awards from IEEE/ACM conferences, journals, and technical committees. His research has been sponsored by RGC, NSFC, ITF, MOST, NRC, JSPS, MIC, JST, and industry. He is a Changjiang Chair professor awarded by the Ministry of Education of China. He was also the chair of organizing and technical committees of numerous international conferences.

Hongyang Chen (M'08-SM'16) received the B.S. and M.S. degrees from Southwest Jiaotong University, Chengdu, China, in 2003 and 2006, respectively, and the Ph.D. degree from The University of Tokyo, Tokyo, Japan, in 2011. He is currently a Senior Research Expert with Zhejiang Lab, China. Currently, he is an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL. He has been selected as the Distinguished Lecturer of the IEEE Communication Society from 2021 to 2022. He is an adjunct professor at Hangzhou Institute for Advanced Study, The University of Chinese Academy of Sciences, China.

Chentao Wu received the BS, ME and PhD degrees in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2004, 2006, and 2010, respectively, and the PhD degree in electrical and computer engineering from Virginia Commonwealth University, Richmond, in 2012. He is a professor at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.

Jie Wu (Fellow, IEEE) received the Ph.D. degree in computer engineering from Florida Atlantic University, Boca Raton, FL, USA, in 1989. He is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He is also the Director of International Affairs at the College of Science and Technology. Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

Wei Zhao (Fellow, IEEE) completed his undergraduate studies in physics at Shaanxi Normal University, China, in 1977, and received his MSc and PhD degrees in Computer and Information Sciences at the University of Massachusetts at Amherst in 1983 and 1986, respectively. Professor Zhao has served important leadership roles in academic including the Chief Research Officer at the American University of Sharjah, the Chair of Academic Council at CAS Shenzhen Institute of Advanced Technology, and the eighth Rector of the University of Macau. Professor Zhao was awarded the Lifelong Achievement Award by the Chinese Association of Science and Technology in 2005.