Long-Term Adaptive VCG Auction Mechanism for Sustainable Federated Learning With Periodical Client Shifting

Leijie Wu, *Student Member, IEEE*, Song Guo, *Fellow, IEEE*, Zicong Hong, *Graduate Student Member, IEEE*, Yi Liu, *Student Member, IEEE*, Wenchao Xu, *Member, IEEE*, and Yufeng Zhan

***Abstract*—Federated Learning (FL) system needs to incentivize clients since they may be reluctant to participate in the resource consuming process. Existing incentive mechanisms fail to construct a sustainable environment for the long-term development of FL system: 1) They seldom focus on system economic properties (e.g., social welfare, individual rationality, and incentive compatibility) to guarantee client attraction. 2) Current online auction modeling methods divide the whole continual process into multiple independent rounds and solve them one-by-one, which breaks the correlation between each round. Besides, the inherent characteristics of FL system (model-agnostic and privacy-sensitive) also prevent it from the optimal strategy by precise mathematical analysis. 3) Current system modelings ignore the practical problem of periodical client shifting, which cannot adaptively update its strategy to handle system dynamics. To overcome the above challenges, this paper proposes a long-term adaptive Vickrey–Clarke–Groves (VCG) auction mechanism for FL system, which incorporate a multi-branch deep reinforcement learning (DRL) algorithm. First, VCG auction is the only one that can simultaneously guarantee all crucial economic properties. Second, we extend the economic properties to long-term forms and apply the experience-driven DRL algorithm to directly obtain long-term optimal strategy, without any prior system knowledge. Third, we reconstruct a multi-branch DRL network to accommodate periodical client shifting by adaptive decision head switching for different time periods. Finally, we theoretically prove he extended economic properties (i.e., IC) and conduct extensive experiments on several real-world datasets. Compared with state-of-the-art approaches, the long-term social welfare of FL system increases by 36% with a 37% reduction in payment. Besides, the multi-branch network can adaptively handle periodical client shifting on the timeline.**

***Index Terms*—Federated learning, deep reinforcement learning, incentive mechanism, game theory.**

I. Introduction

FEDERATED learning (FL) is widely adopted in many practical applications to train a global shared model with the collaboration of decentralized client devices, under the orchestration of a cloud server [1]. It is designed for client data privacy protection, where the local raw data of each client will never be transferred to or shared with others. The cornerstone of FL's success is built on extensive client participation, since obtaining a model with good generalization capability requires training on extensive data samples. Besides, due to the data privacy protection of FL, clients who participate in FL must perform local training to update the model and then push back to the server-side, which consumes significant computation and communication resources on the client-side. Considering the current FL application scenario, clients usually use some resource-constrained mobile devices, such as smartphones, tablets and laptops [2]. Without proper incentive mechanism, they are reluctant to contribute their data and resources to participate in such resource-consuming FL iterations.

Some existing works for FL focus on incentive mechanism to attract clients to participate in FL training. For example, Zhan et al. formulate the problem as a stackelberg game and maximize the server-side utility by rewarding clients according to their contribution and resource conditions [3], [4]. Kang et al. assign different reputation credit to clients based on their contribution and only permit clients with high reputation to participate [5]. Yu et al. propose a fairness-aware mechanism to reward clients based on their heterogeneous data and resource. However, these mechanism designs neglect the following crucial challenges and thus fail to achieve the long-term sustainable development of the FL system. A demonstration is shown in Fig. 1 and more details are presented in Section III to analyze them.

1) Social Welfare Maximization. The sufficient research studies in economics field indicates that only maintaining a sustainable environment can promote the joint development of all parties inside the whole system [6], [7], [8]. To construct a sustainable development FL system, one of the most important

Manuscript received 20 June 2023; revised 28 August 2023; accepted 11 September 2023. Date of publication 19 September 2023; date of current version 4 April 2024. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2021B0101400003, in part by Hong Kong RGC Research Impact Fund under Grants R5060-19 and R5034-18, in part by the Areas of Excellence Scheme under Grant AoE/E-601/22-R, in part by General Research Fund under Grants 152203/20E, 152244/21E, 152169/22E, and 152228/23E, and in part by Shenzhen Science and Technology Innovation Commission under Grant JCYJ20200109142008673. Our previous work related to this paper was presented at the IEEE 42nd International Conference on Distributed Computing Systems (ICDCS 2022), [DOI: 10.1109/ICDCS54860.2022.00091]. Recommended for acceptance by S. Wang. *(Corresponding author: Song Guo.)*

Leijie Wu, Zicong Hong, Yi Liu, and Wenchao Xu are with the Hong Kong Polytechnic University, Hong Kong, SAR, China (e-mail: lei-jie.wu@connect.polyu.hk; zicong.hong@connect.polyu.hk; joeylau.liu@connect.polyu.hk; wenchao.xu@polyu.edu.hk).

Song Guo is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: songguo@cse.ust.hk).

Yufeng Zhan is with Beijing Institute of Technology, Beijing 100811, China (e-mail: yufeng.zhan@bit.edu.cn).

Digital Object Identifier 10.1109/TMC.2023.3317063

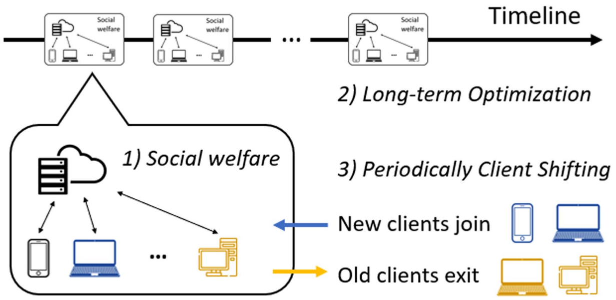


Fig. 1. Challenges for current incentive mechanism design.

economic properties is to maximize the system social welfare, while some additional properties, i.e., incentive compatibility (IC) and individual rationality (IR), can further optimize the sustainability. However, most previous works only focus on server-side utility maximization without considering client utilities. Besides, although a few works also focus on social welfare maximization, but they cannot simultaneously satisfy other economic properties (e.g., IC and IR) [9].

2) Long-term Optimization. Since the FL training process is comprised by a series of consecutive communication rounds. The training results from the current round's decision will have a long-term influence on all subsequent rounds. The previous work adopts online mechanism design for their problem modeling [10], [11], [12]. They decompose the entire process into multiple independent sub-problems in each round and optimizes them respectively, which undermines the long-term correlation among different rounds. Therefore, an incentive mechanism with long-term optimization is necessary for FL system.

3) Periodical Client Shifting. In the current mechanism design, there is a common but rarely studied problem, that is, the periodically client shifting [13], [14]. It has been observed in FL applications where the available client devices periodically change over time, due to the impact of clients from different time zones. All previous works fail to handle this dynamic change, since the optimal mechanism obtained on previous client set cannot be adaptively adjusted. Therefore, the mechanisms employed in actual FL system need to accommodate periodically client shifting in the whole long-term process.

In this paper, to fulfill the missing economic properties and establish a sustainable FL system, we introduce the Vickrey–Clarke–Groves (VCG) auction to guide our mechanism design [15], [16]. We introduce the auction-based mechanism as our backbone for the following motivations: 1) Social welfare maximization is a common concept in auction mechanisms, while other mechanisms (such as Stackelberg Game) mainly focus on unilateral utility maximization. 2) Auction mechanisms also have many other economic properties, such as incentive compatibility (IC) and individual rationality (IR), which is crucial for theoretically constructing a sustainable FL environment. 3) The relationship between clients and the server is equivalent in auction mechanism, where clients can make any offer according to their own needs and the server also has the freedom to refuse the offer from any client. This is more suitable and practical for the FL system in the real world. Besides, compared with all other auction mechanisms, VCG auction is the only one that can simultaneously satisfy both IC and IR properties while maximizing social welfare. However, the inherent privacy protection of FL makes it impossible for the server to obtain the private information of clients, such as computation cost, communication cost, and data-related information. Moreover, considering the black-box of deep neural network and the complex model aggregation, the change process of FL global model performance is impossible to be accurately modeled. The above difficulties invalidate all current auction-based mechanisms, since they cannot obtain precise system modeling for mathematical analysis.

Therefore, to satisfy the requirements of privacy protection, black-box model and long-term optimization, we utilize the deep reinforcement learning (DRL) to obtain the optimal mechanism by a experience-driven and model-free manner. We design a long-term online VCG auction mechanism, which extend these vital economics properties into long-term forms, i.e., long-term social welfare, long-term IC, and long-term IR. Instead of the previous independent optimization based on single-round decomposition, the DRL algorithm directly optimize the long-term social welfare, by continually interacting with the environment, without any prior knowledge of FL system.

Moreover, for the challenge of periodically client shifting, we take the advantages of representation learning, which can learn a shared representation of different tasks [17]. For example, in vision tasks, it can learn a common feature extractor, whereas in language tasks, it can learn shared embeddings and grammars of the context.

In our scenario, we train a multi-head network for DRL agent to address the periodically client shifting, where each head (Decision Generator) is for specific time zone and they share the common environment analyzer (Feature Extractor). The multi-branch DRL strategy is based on the assumption of stable periodically client shifting.

Our contributions are summarized as follows:

- We establish a sustainable FL system by introducing the long-term online VCG auction into problem formulation. It can simultaneously satisfy vital economics properties of extended long-term forms, while maximizing the long-term social welfare.
- We utilize a model-free and experience-driven reinforcement approach to overcome the difficulties of privacy protection, black-box model, and long-term optimization. Moreover, we adopt parameter-based knowledge transfer to speed up the training of different DRL-agents, which reduces the original training cost by half.
- We design a multi-head network to address periodically client shifting. Each head is a decision generator for specific client set in each time zone and the shared body is the environment analyzer for all heads. The head can be adaptively switched according to the client shifting caused by time zone change.
- We provide theoretical analysis to prove our long-term economic properties (i.e., IC) and conduct experiments on different datasets, i.e., MNIST, Fashion-MNIST and CIFAR-10. The experiment results illustrate that we achieve 36% higher long-term social welfare with 37% less payment, comparing with status quo benchmarks. Besides, facing

periodically client shifting, our multi-head network can maintain 30% higher performance than the single network.

II. Related Work

A. Incentive Mechanism for Federated Learning

Since the training process consumes the clients' energy, data, and time, clients will be reluctant to participate in the learning process without receiving compensation. Therefore, an incentive mechanism design is critical for FL to motivate the clients to contribute their resources [18]. In the following, we conclude a few existing works for FL incentive mechanisms.

First, some works distribute the money to the clients heuristically. For example, Weng et al. reward clients based on their processed data quantities and honest behaviours [19]. Bao et al. distribute the profit to the clients according to their reliability and contribution (decided by the remarks of the other clients and trained model buyers) [20].

Second, some works adopt a non-cooperative game in which FL servers or users aim to maximize their unilateral utility for FL incentive mechanism design. For example, some works present a Stackelberg game-based FL incentive mechanism for a Nash equilibrium (NE) between the FL server and clients [21], [22]. To solve the difficulties of modelling raised by the information isolation in FL, Zhan et al. propose an experience-driven DRL mechanism for the optimal pricing strategy of servers to achieve efficient learning in each round [3], [4]. Moreover, to overcome the challenge of myopia optimization existed in the previous works, Liu et al. design an incentive-driven long-term mechanism for FL based on hierarchical DRL [23].

Next, because auction theory is an efficient mathematical tool for pricing and allocation [15], some works study auction-based incentive schemes in FL. For example, Zeng et al. propose a multi-dimensional procurement auctionbased incentive mechanism named Fmore to encourage more high-quality edge nodes with low cost to participate in FL training [24]. Jiao et al. propose a reverse multi-dimensional auction that chooses the clients and decides the rewards in a randomized and greedy manner for wireless FL service market [25]. online auction for efficient FL client selection strategy with energy constraint [26]. Deng et al. propose a quality-aware reverse auction-based incentive mechanism to maximize the sum of the estimated learning qualities of clients [27]. Tang et al. design an incentive mechanism to achieve not only social welfare maximization but also individual rationality and budget balance in cross-silo FL [28]. A few of the existing works also focus on the social welfare maximization of FL system. Zhou et al. apply the greedy algorithm for FL social welfare maximization problems, while guaranteeing critical economic properties [9]. However, they don't consider the long-term influence of the FL system (i.e., the decision made in the current FL round is not independent in every round, which will influence all subsequent rounds), as well as the information isolation and model unknown constraints.

Furthermore, the long-term influence of FL and its optimization problem also received wide attention from many existing works [23], [29]. Liu et al. design a reinforcement-based incentive mechanism to achieve the long-term optimization of server-side utility [23]. Xu et al. realize that FL rounds are not independent but have continual influences on the final results, and thus design a client selection method with long-term optimization target [29].

In comparison with our work, the existing works have three major disadvantages as follows. First, most of the existing works cannot simultaneously satisfy the economic properties including social welfare, incentive compatibility, and individual rationality. These economic properties are the key elements for long-term FL training and a sustainable FL system. Second, the FL training is composed of multiple successive rounds, but most of the existing works decompose the whole FL training process into a series of independent sub-problems, each corresponding to a round. Thus, these FL incentive mechanisms can only solve the problem in a myopic manner. Third, the performance of most of the existing works depends on precise system modeling for the dynamic FL training process. However, the inherent privacy protection and the black box of DNN models in FL make such precise modeling impractical.

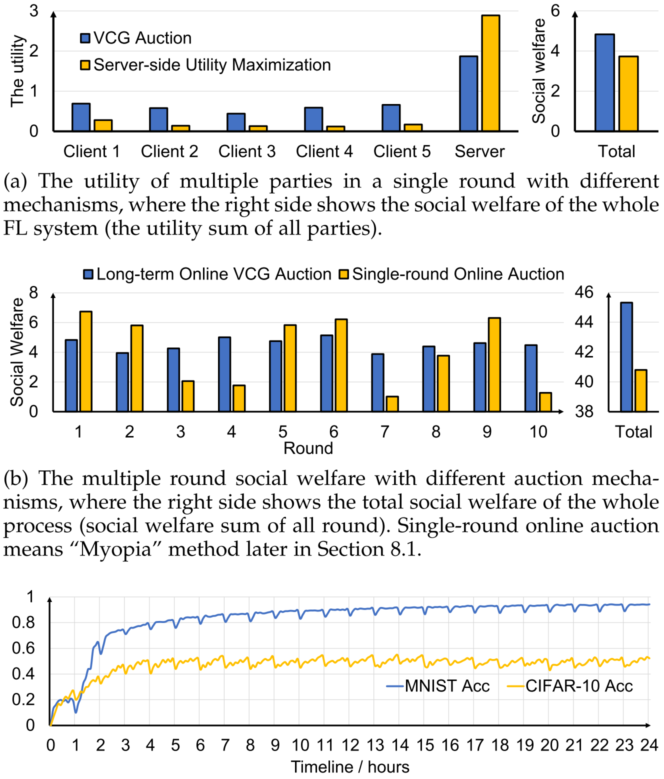
B. Learning-Based Incentive Mechanism

Motivated by the potential of machine learning to optimize the solutions to complex problems, plenty of recent works leverage learning-based techniques for optimal incentive mechanism design, particularly auction mechanisms. For example, Dutting et al. model the automated design of optimal auctions as a constrained learning problem of a multi-layer neural network [30], [31]. Rahme et al. refine Dutting et al.'s work based on two recent results from machine learning (i.e., amortization) and theoretical auction design (i.e., stationary Lagrangian) [32]. Feng et al. study the design of optimal auctions for settings with private budgets [33] and Golowich et al. study that for multi-facility setting [34]. Based on the deep learning technique, Luong et al. design an optimal auction for the edge resource allocation in mobile blockchain networks [35]. Liu et al. propose a learning-based end-to-end auction mechanism for industrial e-commerce advertising. [36]. Besides, some learning-based mechanism are based on reinforcement learning (RL) for online mechanism design. For example, Tang et al. [37] and Cai et al. [38] develop a RL-based mechanism design framework to design and optimize mechanisms for e-commerce. Zhang et al. propose a generalized second-price auction mechanism based on a model-free RL algorithm [39].

In comparison with our work, most of the existing works adopt learning techniques for a single-round optimal auction. As discussed in Section III, such a design performs poorly in a FL system with multiple consecutive rounds. Although a few works [37], [38], [39] adopt RL for an online mechanism design, they are designed for e-commerce rather than FL and thus cannot achieve the same goals of FL incentive mechanisms as described in Section II-A. Different from the existing learning-based incentive mechanism, this paper aims to adopt DRL approach to design a long-term optimal online auction mechanism for FL.

III. Preliminary Analysis

To fully understanding the challenges that cannot be solved by existing work, we conduct extensive pre-experiments to analyze and illustrate them as follows.



(a) The utility of multiple parties in a single round with different mechanisms, where the right side shows the social welfare of the whole FL system (the utility sum of all parties).

(b) The multiple round social welfare with different auction mechanisms, where the right side shows the total social welfare of the whole process (social welfare sum of all round). Single-round online auction means "Myopia" method later in Section 8.1.

(c) The influence of periodically client shifting for the global model accuracy change in FL system, where we set the available client subset to change at each whole hour in the day.

Fig. 2. Pre-experiment results to illustrate the critical challenges that existing mechanism cannot handle, including Social Welfare Maximization, Long-term Optimization and Periodically Client Shifting.

1) Social Welfare Maximization: The majority of existing incentive mechanisms are mainly focus on server-side utility maximization, they adopt different strategies to attract client participation, by elaborate reward designs based on resource conditions [3], [4], [40] or reputation credit allocation [5]. However, FL is a complex system with multi-party collaboration, a selfishly mechanism that only maximize the unilateral server-side utility will undermine client participation. Extensive studies in economics have illustrated that only a sustainable environment can promote the co-development of the whole system, and social welfare maximization is one of the most important properties for sustainability. Our pre-experiment results in Fig. 2(a) show the utilities of all participants (both clients and sever) under different mechanisms in a single round. For existing server-side utility maximization methods, the most system utility is concentrated on the server-side by squeezing all clients, which will significantly undermines client involvement. On the contrary, by actively sharing a portion of its utility with clients, the server can strongly motivate clients and achieve a higher system social welfare under our mechanism (VCG auction).

2) Long-Term Optimization: Since the FL training process is comprised of multiple consecutive rounds, the online auction formulation [10], [11], [12] is appropriate for the system modeling of our social welfare maximization. However, all existing online auctions adopt similar solutions by splitting the entire process into a series of independent sub-problems in each single round and solving them one-by-one. This splitting method destroys the inherent before-and-after correlation in FL, because the global aggregation model in the previous round will serve as a new start-point for the next round. The optimal mechanism must consider its long-term subsequent influence when making decision for current round. Our pre-experiment results in Fig. 2(b) also empirically illustrate that such splitting degrades the total system social welfare from a long-term perspective. Although our long-term mechanism may not be optimal in some particular rounds (i.e., round 5 and 6), it can guarantee the long-term optimality for the entire process, by considering its subsequent influence and sacrificing a portion of current social welfare to promote future improvement.

3) Periodically Client Shifting: It is observed from practical FL applications that the available client population has a periodical change over time due to different time zones [13], [14]. However, all existing mechanism designs cannot deal with this system dynamic because their strategies are static, in other word, the optimal strategy derived from previous client population will fail in next period with changes. The pre-experiment results in Fig. 2(c) demonstrate the tendency of global model accuracy changes with different datasets. If the mechanism cannot perceive the impact of client shifting to adaptively update its strategy, the previous strategy will fail to handle the new client subset, which also leads to the social welfare reduction in the future (See experiment results in Fig. 9).

IV. System Model & Problem Formulation

A. Federated Learning System

The federated learning framework is composed by a central server and a set of clients denoted by $\mathcal{N} = \{1, \ldots, N\}$. Each client $i \in \mathcal{N}$ has a local dataset $\mathcal{D}_i$ composed by multiple data samples $\{\boldsymbol{x}_j, y_j\}_{j\in\mathcal{D}_i}$, and perform local model training with a pre-defined loss function f. Specifically, in t-th round, only a subset of total client population $\mathcal{G}^t \subseteq \mathcal{N}$ is available to connect with the server due to periodically client shifting. All connected clients will receive a global model $\boldsymbol{\omega}^t$ from server and train their own local model by its local loss on their own dataset $\mathcal{D}_i$, where $d_i = |\mathcal{D}_i|$ denotes the sample size of client i's local dataset. Then, clients will send the updated local model $\boldsymbol{\omega}_i^t$ back to server for model aggregation:

$$\boldsymbol{\omega}^{t+1} = \sum_{i=1}^{|\mathcal{G}^t|} \frac{d_i}{d} \boldsymbol{\omega}_i^t, \quad i \in \mathcal{G}^t, \tag{1}$$

where $d = \sum_{i=1}^{|\mathcal{G}^t|} d_i$ is the total data size of t-th round. Finally, the new global model $\boldsymbol{\omega}^{t+1}$ will serve as starting point for the next round $t+1$. The above process will repeat until the model converges or meets specified requirements. Normally, the server has a validation dataset $\mathcal{D}_v$ and the target is to obtain the optimal global model $\boldsymbol{\omega}^*$ that

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} F(\boldsymbol{\omega}), \text{ where } F(\boldsymbol{\omega}) = \frac{1}{|\mathcal{D}_v|} \sum_{j\in\mathcal{D}_v} f(j, \boldsymbol{\omega}). \tag{2}$$

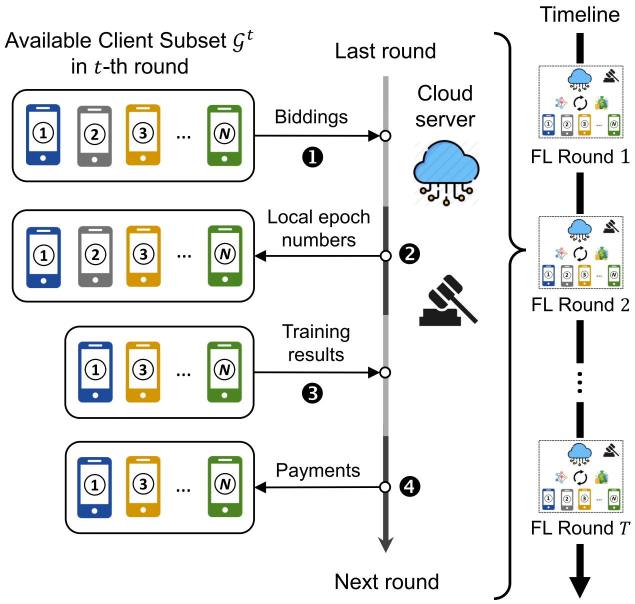


Fig. 3. Illustration of online auction-based FL system modeling.

Generally speaking, as the loss $F(\omega)$ decreases, the accuracy $A(\omega)$ will gradually increase. Thus, we adopt the model accuracy $A(\omega)$ on validation dataset $\mathcal{D}_v$ as the metric to measure model performance in the following part.

B. Auction-Based Modeling for FL System

To apply the VCG auction mechanism on our federated system, we built the system model based on typical online auction as shown in Fig. 3, where the whole process is comprised by a series of consecutive communication rounds and each round corresponds to a sub-auction. More specifically, in each round $t \in [1, T]$, for those available clients $i \in \mathcal{G}^t$ of current round, they will first provide their own bidding price b_i^t for joining one local epoch training to the server, which is based on each client's true value v_i^t (Fig. 3-❶). Then, the server will determine its training strategy of those available clients in current round as $\boldsymbol{\psi}^t = \{\psi_1^t, \ldots, \psi_{|\mathcal{G}^t|}^t\}$, where ψ_i^t denotes the local epoch training number of client i in t-th round (Fig. 3-❷). Next, each client will perform their specific training task ψ_i^t and upload the training results (local models) back to the server (Fig. 3-❸). Finally, the server will compute the payment p_i^t to each client based on their contribution with VCG payment rule (Fig. 3-❹). The details of above each step will be elaborated as follows.

❶ *Client Bidding*: For each available client $i \in \mathcal{G}^t$ of t-th round, the bidding price b_i^t provided by them is based on its true value v_i^t. The true value is an estimation of their various resources consumption after participating in current FL training, which may vary significantly among clients due to their heterogeneous hardware configurations and Non-IID data distributions. The main components of true value are listed as follows:

1) Computation Energy Cost. Since the data type of each client's local dataset is the same, i.e., all MNIST data samples are 28 × 28 pixels gray-scale images, the number of CPU cycles required to execute a data sample is the same for all clients, which is denoted by c and can be obtained in advance. Let δ_i denote the CPU-cycle frequency of client i to execute the local training task and β_i denote the effective capacitance coefficient of client i's computation chip-set, where β_i and δ_i of each client i can vary significantly due to the heterogeneous hardware configurations. Recall that $d_i = |\mathcal{D}_i|$ is the local dataset size of client i. According to the widely accepted system energy model [41], the computation energy cost of client i to perform one local epoch training is

$$e_i^{comp} = \beta_i c d_i \delta_i^2. \tag{3}$$

2) Communication Energy Cost. In a traditional FL system, the local model structure of all clients must be the same to support the global model aggregation, thus the required communication volume of each client to upload their local models is the same, which is denoted by ξ. Let B_i^t denote the communication bandwidth of client i in t-th round. The communication time of client i can be computed by $T_i^{t,comm} = \xi / B_i^t$. According to the typical energy modeling [41], the communication energy cost of client i in t-th round is defined as:

$$E_i^{t,comm} = \epsilon_i T_i^{t,comm} = \frac{\epsilon_i \xi}{B_i^t}, \tag{4}$$

where ϵ_i is the unit energy consumption of client i for uploading the local model and it varies greatly depending on heterogeneous hardware configuration.

3) Data Usage. In addition to the above energy cost, extensive studies [42], [43], [44] also point out that the local dataset is another valuable resource of each client, which need to be charged according to its usage during participating in local training. However, due to the data Non-IID distribution of clients, the data value of each client is dynamic depending on the data demand of server in different training stage. For example, to improve the global model performance in current round, the server will select those clients with the largest likelihood of having good quality data according to its historical experience. In other words, if a client is more frequently selected by the server in previous rounds (assigned more local epoch number), it has a higher data value to help the server improve global model performance. To formulate the dynamic changes of client data value in different training stages, we introduce the economic model of supply and demand [45], where the client is the data supply side and the server is the data demand side. Assume that the data value of client i in t-th round is defined as ρ_i^t according to the marketing rule [46], we can formulate the dynamics of data value in the market as $\rho_i^t = M(\rho_i^{t-1}, \psi_1^{t-1})$, where M is a Markovian function reflecting how the data demand influences the data value [47]. We construct the Markovian function by comparing ψ_i^t and $\bar{\psi}_i$, where $\bar{\psi}_i$ is the historical average local epoch number assigned to client i in previous rounds. If the local epoch number ψ_i^t determined by server is larger than history average $\bar{\psi}_i$, then the system is a supply-side market in current t-th round and client i will consider to increase its data value by $(\psi_i^t - \bar{\psi}_i)/\bar{\psi}_i$, and vice versa.

Based on above three components, each client first estimates its true value of current round by:

$$\bar{v}_i^t(\bar{\psi}_i^t) = \overbrace{\eta_1 * \underbrace{(e_i^{comp} \times \quad \bar{\psi}_i^t)}_{per\ local\ epoch} + \eta_2 * E_i^{t,comm} + \eta_3 * \rho_i^t}^{per\ round}, \quad (5)$$

where η_1, η_2, η_3 are the hyperparameters for preference adjustment. Since the exact local epoch number of current round is determined by the server later, which is unknown in client bidding stage, the true value estimation is depending on the historical average local epoch number $\bar{\psi}_i^t$ of previous rounds. The computation energy cost e_i^{comp} is evaluated for each local epoch. The communication energy cost $E_i^{t,comm}$ and data value ρ_i^t are evaluated for the whole round, because each local training round only needs to utilize the data once and upload the model once.

Finally, due to the nature of selfishness and rationality, client may raise a bidding price b_i^t different from its true value estimation v_i^t to strive more payment from the server. Therefore, the bidding price b_i^t provided by client i in t-th round based on its true value estimation is defined as:

$$b_i^t\left(\bar{\psi}_i^t\right) = v_i^t\left(\bar{\psi}_i^t\right) + |N\left(\mu, \sigma^2\right)|, \quad (6)$$

where $N(\mu, \sigma^2)$ is a Gaussian noise with $\mu = 0$ and $\sigma = 1$.

❷ *Server Strategy Determination*: After receiving all client bidding prices $\boldsymbol{b}^t = \{b_1^t, \ldots, b_{|\mathcal{G}^t|}^t\}$, the server will feed them into of a DRL strategy network parameterized by θ as the input, and the network output $\boldsymbol{\psi}^t = \{\psi_1^t, \ldots, \psi_{|\mathcal{G}^t|}^t\}$ is the local epoch number strategy for all clients in t-th round. Please note that there is a voluntary two-way selection between the server and clients, the server can refuse the participation of specific clients by assigning a local epoch number of zero, and clients can deliberately provide unreasonable bidding prices (or not provide) to force the server to give it up.

❸ *Training Result Aggregation*: Each client will conduct its own local training task according to the required local epoch number b_i^t from the server, and then upload the updated local model back to server-side. Besides, the client true value ρ_i^t will also be updated based on $\rho_i^t = M(\rho_i^{t-1}, \psi_1^{t-1})$.

❹ *Payment to Client*: The server will allocate the corresponding payment p_i^t to each client according to the VCG payment rules defined later in Section V-C.

After all auction process of current round are completed, the server and clients will compute their respective utilities and update their strategies for next round. Based on the payment p_i^t from the server, the utility of client i in t-th round is computed as:

$$\begin{aligned} U_i^t(\psi_i^t) &= p_i^t - v_i^t\left(\psi_i^t\right) \\ &= p_i^t - \left(\eta_1 e_i^{comp}\psi_i^t + \eta_2 E_i^{t,comm} + \eta_3 \rho_i^t\right), \end{aligned} \quad (7)$$

The hyper-parameters η_1, η_2, and η_3 here have no special meaning, which are some preference adjustment weights. For example, if we assign a large value for eta_1, it means that the current client i is more concerned about its computation cost when evaluating its own utility function, and vice versa for other hyper-parameters.

TABLE I
PARAMETER NOTATIONS

Parameter	Definition
$t \in T$	Round index t and Total FL communication round T
$\mathcal{N}$ & N	Total client set & Total client number
$\boldsymbol{\omega}^t$	FL global model in t-th training round
$\mathcal{G}^t$	Client subset that join in the t-th round FL training
$\mathcal{D}_i$ & d_i	Local dataset & local data size of client i
A^t	Accuracy of FL global model in t-th round
b_i^t & v_i^t	Bidding price & true value of client i in t-th round
ψ_i^t	Local epoch training number of client i in t-th round
p_i^t	Payment to client i in t-th round
c	Required CPU-cycle to conduct one data sample
δ_i	CPU-cycle frequency of client i
β_i	Effective capacitance coefficient of client i′s chip-set
e_i^{comp}	Computation energy cost of client i in one local epoch
ξ	Communication volume of FL model
B_i^t	Communication bandwidth of of client i in t-th round
ϵ_i	Unit communication energy cost of client i
$E_i^{t,comm}$	Communication energy cost of client i in t-th round
ρ_i^t	Data value of client i in t-th round
U_i^t & U_s^t	Utility of client i & server in t-th round
U_i & U_s	Long-term utility of client i & server
η_1, η_2, η_3	Weights of client cost estimation
λ	Preference adjustment parameter of social welfare
S^t	Social welfare of t-th round
S	Long-term social welfare of FL system
ψ^*	Optimal long-term local epoch number strategy
$k \in K$	Client set index k and Total client set number K
$\boldsymbol{\theta}_f$	Shared feature extractor layers of multi-branch model
$\boldsymbol{\theta}_k$	Specialized decision head for k-th client set
τ	Client distribution similarity

According to the typical assumption in economics that each client is individual rational, the target of client i is to maximize its long-term utility during the entire FL training process, which is $U_i = \sum_{t=1}^{T} U_i^t$.

For the server-side, after the global model aggregation, the server can obtain the global model accuracy $A(\cdot)$ of current t-th round on its validation dataset $\mathcal{D}_v$ with Eq. (2). Therefore, the utility of server in t-th round is computed by:

$$U_s^t\left(\boldsymbol{\psi}^t\right) = \lambda \cdot \Delta A^t\left(\boldsymbol{\psi}^t\right) - \sum_{i=1}^{|\mathcal{G}^t|} p_i^t, \quad (8)$$

where $\Delta A^t(\boldsymbol{\psi}^t) = A(\boldsymbol{\omega}^t) - A(\boldsymbol{\omega}^{t-1})$ is the global model accuracy increment in t-th round after adopting the local epoch number strategy $\boldsymbol{\psi}^t$, $\mathcal{G}^t$ is the available client subset in t-th round, and λ is a non-negative parameter to adjust the server preference. A large λ means the server can afford more payments to facilitate clients participation, in exchange for model accuracy improvement, vice versa. All parameter notations appeared in this paper are summarized in Table I for reference.

From above auction modeling of FL system, we can observe that the server strategy $\boldsymbol{\psi}^t$ will lead to two impacts: 1) Since Non-IID data distribution of clients, the decision of $\boldsymbol{\psi}^t$ will influence the model accuracy improvement by selecting different client combination to join in FL training, which can be regarded as a common client selection problem. 2) $\boldsymbol{\psi}^t$ will also influence the FL training cost, including computation & communication energy cost and data usage. Therefore, the target of our mechanism is to find the optimal trade-off between the system performance improvement and training cost by adaptively applying different local epoch number strategy in each round.

C. Problem Formulation

Before giving our problem formulation, we introduce several traditional economic properties that should be guaranteed in a well-designed auction mechanism [15]:

- *Social welfare:* the utility sum of all participants within the system, including both the server and clients.
- *Incentive compatibility:* all participants can obtain the best return, if and only if they behave truthfully (truthful bidding price for clients and truthful payments for the server).
- *Individual rationality:* all participants can obtain non-negative utility.

However, the above traditional economic properties can only be guaranteed in a single auction, which is unsuitable for our long-term online auction modeling, since the whole FL training process is comprised by a series of consecutive communication rounds and each round is formulated as one auction. To further enhance the mechanism reliability, we extend all economic properties into a novel long-term form, which can be perfectly adapted to our long-term requirements. Therefore, the final target of our mechanism is to achieve long-term social welfare maximization of a FL system, while guaranteeing all other critical economic properties from a long-term perspective. We describe the details of these long-term economic properties as below:

1) **Long-term social welfare maximization**: Social welfare is a widely accepted metric to evaluate the performance of an auction mechanism [10], [11], [12], [15]. According to the utility definitions of the server and clients in Section IV-B, we can compute the social welfare of t-th round as:

$$\begin{aligned} S^t\left(\boldsymbol{\psi}^t\right) &= U_s^t\left(\boldsymbol{\psi}^t\right) + \sum_{i=1}^{|\mathcal{G}^t|} U_i^t\left(\boldsymbol{\psi}^t\right) \\ &= \lambda \cdot \Delta A^t\left(\boldsymbol{\psi}^t\right) \\ &\quad - \sum_{i=1}^{|\mathcal{G}^t|}\left(\eta_1 e_i^{comp}\psi_i^t + \eta_2 E_i^{t,comm} + \eta_3 \rho_i^t\right), \end{aligned} \tag{9}$$

where $\mathcal{G}^t$ is the available client subset in t-th round. Furthermore, the long-term social welfare is defined as the utility sum of both the server and clients in all rounds:

$$\begin{aligned} S(\boldsymbol{\psi}) &= \sum_{t=1}^{T} U_s^t\left(\boldsymbol{\psi}^t\right) + \sum_{t=1}^{T}\sum_{i=1}^{|\mathcal{G}^t|} U_i^t\left(\boldsymbol{\psi}^t\right) \\ &= \sum_{t=1}^{T}\left(U_s^t\left(\boldsymbol{\psi}^t\right) + \sum_{i=1}^{|\mathcal{G}^t|} U_i^t(\boldsymbol{\psi}^t)\right) = \sum_{t=1}^{T} S^t\left(\boldsymbol{\psi}^t\right). \end{aligned} \tag{10}$$

Therefore, our target is to find the optimal long-term local epoch number strategy $\boldsymbol{\psi}^*$ that can maximize the long-term social welfare, i.e.,

$$\boldsymbol{\psi}^* = \arg\max S\left(\boldsymbol{\psi}\right). \tag{11}$$

2) **Long-term incentive compatibility**: When an auction satisfies incentive-compatible (truthful), it means that any client i in t-th round will obtain the best utility gain, if and only if it announces a bid that truthfully reveals its true value v_i^t. Assume that $\boldsymbol{\psi}^*$ is the long-term optimal strategy when client i bids truthfully in the whole process and $\tilde{\boldsymbol{\psi}}^*$ is the long-term optimal strategy when it bids untruthfully. Then, the definition of long-term incentive compatibility is:

$$U_i\left(\boldsymbol{\psi}^*\right) \geq U_i\left(\tilde{\boldsymbol{\psi}}^*\right), \forall i \in \mathcal{N}, \tag{12}$$

where $U_i = \sum_{t=1}^{T} U_i^t$ is the long-term utility sum of client i in the whole process.

3) **Long-term individual rationality**: Individual rationality means that each client will choose to participate in the FL training only when they can obtain a positive utility. Therefore, to maintain attraction to clients in whole process, it must satisfy long-term individual rationality as:

$$U_i\left(\boldsymbol{\psi}^*\right) > 0, \forall i \in \mathcal{N}. \tag{13}$$

Notice: All previously described long-term economic properties are guaranteed from the long-term perspective, which means they may not be satisfied in every FL round. In other words, the server or clients can accept lower or even negative utility gain in some specific rounds to further promote higher long-term utility maximization for the whole process, which is also illustrated in Fig. 2.

V. Incentive Mechanism Design

A. Motivation: Why Must DRL

Since VCG-based auction is the only one that can simultaneously satisfy all above mentioned economic properties [16], its an ideal auction type adopted into our mechanism. The standard VCG auction is consist of two steps: First, it requires to obtain the optimal strategy by solving the social welfare maximization problem, which are (10) and (11) in our paper. Second, the payment to each client is computed based on the externality that client exerts on other clients, which will be elaborated in Section V-C later. However, directly applying standard VCG auction in our mechanism is difficult, because the following critical challenges prevent us from obtaining the long-term optimal strategy $\boldsymbol{\psi}^*$ by precise mathematical analytical solutions:

1) *Myopic Observation:* To derive the optimal long-term strategy for (10) and (11) through precise mathematical analytical solutions, we have to obtain the complete knowledge about the FL system over its entire future lifespan in advance. It is obviously impossible that we only have the historical knowledge.

2) *Information Isolation:* During the theoretical derivation process of $\boldsymbol{\psi}^*$ for (11), we must substitute (3)-(5) into (9) to obtain the social welfare $S^t(\boldsymbol{\psi}^t)$ of t-th round. However, those client private parameters (e.g., $\beta_i, c_i, d_i, \delta_i, \epsilon_i, B_i^t$) in (3)-(5) is unavailable to the server due to the inherent privacy protection in FL.

3) *Model Unknown:* Due to the black-box of deep neural network and the collaborative training manner in FL system, it is impossible to derive an analytical model of global model performance, i.e., $A(\omega^t)$. In other words, we cannot use mathematical analysis to know the global model accuracy in advance, but only after the actual FL training process is completed.

Different from mathematical analytical solutions, DRL is well suited to our mechanism because of the following reasons: First, as an experience-driven approach, the update of DRL strategy only need to interact with the FL system and does not rely on any prior knowledge inside the FL system, which makes it overcomes the *model unknown* and *information isolation* challenges. Second, the optimization objective of DRL strategy is the expected long-term accumulated reward (i.e., the long-term social welfare optimization in our problem), which assists our mechanism to overcome the *myopic observation* challenge.

Furthermore, previous studies [48] show that most machine learning applications are employed in dynamic environments, where data patterns change over time. Therefore, the server-side global model in FL requires constant updates to adapt to the environment dynamics. During this process, extensive client interaction experience is accumulated in server-side, which also provide conductive precondition for DRL deployment. Motivated by these advantages, we adopt DRL into our long-term online VCG auction mechanism.

B. DRL Design

According to the system model in Fig. 3, we combine the the Markov Decision Process (MDP) with the FL training process, where each round is corresponding to a state transition process $\{s_t, a_t, r_t, s_{t+1}\}$. The traditional Proximal Policy Optimization (PPO) is adopted as our DRL optimization algorithm and other details of our DRL design are described as follows:

State Design: The state s_t of agent in t-th FL round is the bidding price vector of all clients for one local epoch and current round index, which is denoted by $s_t = \{b_1^t, \ldots, b_i^t, \ldots, b_{|\mathcal{G}^t|}^t, t\}$ (see Section IV-B.)

Action Design: The action a_t of agent in t-th FL training round is server's local epoch number strategy for all clients, which is denoted as $a_t = \{\psi_1^t, \ldots, \psi_i^t, \ldots, \psi_{|\mathcal{G}^t|}^t\}$.

Reward Design: The reward r_t of agent is the social welfare of system in t-th FL round, which is represented by $r_t = S^t = \lambda \cdot \Delta A^t - \sum_{i \in \mathcal{G}^t} (\eta_1 e_i^{comp} \psi_i^t + \eta_2 E_i^{t,comm} + \eta_3 \rho_i^t)$. To achieve the long-term optimization objective, we adopt the Monte-Carlo update rule for the DRL algorithm, which only update its strategy in the last round, although the agent can receive the reward every round.

State Transition: In the beginning of t-th round, all clients will provide their bidding price (state s_t) for one local epoch training to the server. Then, after receiving all client states, the server will determine the local epoch number of them (action a_t) according to its DRL strategy network $\pi_\theta(a_t|s_t)$. Next, clients will perform the corresponding local training task based on the requirements (action a_t) and upload their training results back to the server. After that, the server will conduct global aggregation to generate new global model and compute the social welfare of t-th round (reward r_t). Finally, all clients will receive the payments from the server and update their strategy to provide new bidding price for next $t+1$ round (new state s_{t+1}).

C. VCG-Based Payment Rule

Let $S_{\mathcal{N}\setminus i}(\boldsymbol{\psi}^*)$ denote the social welfare after applying the optimal global strategy $\boldsymbol{\psi}^*$, but without considering the cost of client i. Besides, we define $\boldsymbol{\psi}_{-i}^*$ as the optimal individual strategy from a sub-auction, which is also the individual environment that excludes client i.

Then, according to the VCG payment rule, the payment to client i is given as

$$p_i = S_{\mathcal{N}\setminus i}\left(\boldsymbol{\psi}^*\right) - S\left(\boldsymbol{\psi}_{-i}^*\right), \tag{14}$$

where $p_i = \sum_{t=1}^{T} p_i^t$ is the total payment of client i throughout the whole FL training. To implement the above VCG payment rule, we redesign the auction process in Section IV-B by a bookkeeping scheme, which means the contribution of clients in each round will be recorded and each client is paid at the end of the FL process.

Next, we can get the following theorem.

Theorem 1. The reinforcement-based online auction mechanism that produces strategy $\boldsymbol{\psi}$ and payments $\boldsymbol{p}$, is both incentive-compatible and individual rationality.

Proof. Based on the VCG payment rule in (14), we can get

$$\begin{aligned} U_i(\boldsymbol{\psi}) &= p_i - v_i(\boldsymbol{\psi}) \\ &= p_i - \sum_{t=1}^{T}\left(\eta_1 e_i^{comp}\psi_i^t + \eta_2 E_i^{t,comm} + \eta_3\rho_i^t\right) \\ &= S_{\mathcal{N}\setminus i}\left(\boldsymbol{\psi}^*\right) - S\left(\boldsymbol{\psi}_{-i}^*\right) \\ &\quad - \sum_{t=1}^{T}\left(\eta_1 e_i^{comp}\psi_i^t + \eta_2 E_i^{t,comm} + \eta_3\rho_i^t\right) \\ &\geq S\left(\boldsymbol{\psi}^*\right) - S\left(\boldsymbol{\psi}_{-i}^*\right) \geq 0 \end{aligned} \tag{15}$$

Therefore, the individual rationality property of the reinforcement-based online auction mechanism is satisfied.

Next, to prove the truthfulness of our mechanism, we compare the utility of client i under the truthful bidding and an untruthful bidding. Assume that in t-th round, client i reports an untruthful bid b_i^t which is not equal to its true value v_i^t, i.e., $b_i^t \neq v_i^t$, then the optimal strategy becomes $\tilde{\boldsymbol{\psi}}^*$. Its utility under untruthful bidding can be calculated by

$$U_i\left(\tilde{\boldsymbol{\psi}^*}\right) = \left(\tilde{S}_{\mathcal{N}\setminus i}\left(\tilde{\boldsymbol{\psi}^*}\right) - S\left(\boldsymbol{\psi}_{-i}^*\right)\right) - v_i(\boldsymbol{\psi}). \tag{16}$$

Then, the difference of client utilities under truthful and untruthful bidding is

$$\begin{aligned} U_i(\boldsymbol{\psi}) - U_i\left(\tilde{\boldsymbol{\psi}}^*\right) \geq\ & \left(S\left(\boldsymbol{\psi}^*\right) - S\left(\boldsymbol{\psi}^*_{-i}\right)\right) \\ & - \left[\left(\widetilde{S}_{\mathcal{N}\setminus i}\left(\tilde{\boldsymbol{\psi}}^*\right) - S\left(\boldsymbol{\psi}^*_{-i}\right)\right) - v_i(\boldsymbol{\psi})\right] \\ =\ & S\left(\boldsymbol{\psi}^*\right) - \left(\widetilde{S}_{\mathcal{N}\setminus i}\left(\tilde{\boldsymbol{\psi}}^*\right) - v_i(\boldsymbol{\psi})\right) \\ =\ & S\left(\boldsymbol{\psi}^*\right) - \left(S_{\mathcal{N}\setminus i}\left(\tilde{\boldsymbol{\psi}}^*\right) - v_i(\boldsymbol{\psi})\right) \\ \geq\ & S\left(\boldsymbol{\psi}^*\right) - S\left(\tilde{\boldsymbol{\psi}}^*\right). \end{aligned} \tag{17}$$

Since $\boldsymbol{\psi}^*$ maximizes the long-term social welfare of FL system, we can get $U_i(\boldsymbol{\psi}) - U_i(\tilde{\boldsymbol{\psi}}) \geq 0$, which means client i cannot increase its utility by bidding untruthfully. □

VI. Multi-Branch Network for Client Periodically Shifting

The client periodically shifting problem in practical FL applications imposes more stringent requirements on dynamic adaptability for all current incentive mechanism designs. Our traditional single model-based DRL strategy network is unable to handle this dynamic problem, once the available client population is changed, the strategy network trained on the previous environment cannot be adapted to the new environment, which leads to repeatedly retraining a new strategy network for incentive mechanism from scratch. Therefore, we propose to jointly train a multi-branch strategy network to sense client population changes and automatically switch to the corresponding specialized branches on different modes. The multi-branch DRL strategy is based on the assumption of stable periodically client shifting.

A. Multi-Branch DRL Strategy Network

Eichner et al. [13] first illustrated the advantages of training separate models for different client data distributions in FL, when the dynamic change can be sensed during evaluation. However, training and maintaining multiple separate models on the server-side is too resource-consuming in real-world applications. Motivated by studies in representation learning that heterogeneous client data distribution in the same task may share a common representation [49]. For example, in vision tasks, the former part of model (i.e., feature extractor layers) will generate common (low-dimensional) representations for different images, while for language tasks it learns shared word embeddings and grammars from different contexts.

To handle the client periodically shifting problem while learning shared feature representations in a more efficient manner, we adopt the parameter-sharing strategy from multi-task learning [50] to collaboratively train a multi-branch strategy network for our DRL agent. The multi-branch strategy network $\boldsymbol{\theta}$ is comprised of two parts: 1) shared feature extractor layers $\boldsymbol{\theta}_f$ that extract common (low-dimensional) representations, and 2) several following specialized decision heads $\boldsymbol{\theta}_k$ for different client sets in different time zones. The architecture of our multi-branch strategy network is demonstrated in Fig. 4. With the change of time zone, the available client set is periodically shifted. For client set in k-th time zone, the specialized model branch responsible for the strategic decision of incentive mechanism is $\boldsymbol{\theta} = (\boldsymbol{\theta}_f, \boldsymbol{\theta}_k)$.

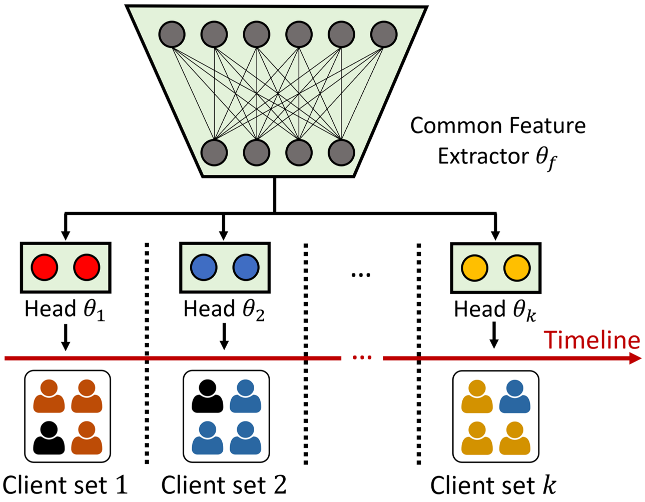


Fig. 4. Architecture of multi-branch strategy network. The available client set is periodically changed in different time zones, which is indicated by different client colors. The model branch responsible for the k-th client set is $\boldsymbol{\theta} = (\boldsymbol{\theta}_f, \boldsymbol{\theta}_k)$. The feature extractor $\boldsymbol{\theta}_f$ is shared by all model branches to provide common feature representations, while each decision head $\boldsymbol{\theta}_k$ is specialized for different client sets.

Algorithm 1: The Training Workflow of Multi-Branch DRL Strategy Network.

```
Input: Total client set $\mathcal{N}$, Initial multi-branch network
  $\boldsymbol{\theta} = (\boldsymbol{\theta}_f, \boldsymbol{\theta}_k), k \in [1, K]$
Output: Optimal multi-branch strategy network $\boldsymbol{\theta}(\psi^*)$.
Initialize the multi-branch network $(\boldsymbol{\theta}_f, \boldsymbol{\theta}_k), k \in [1, K]$.
for time zone $k = 1, 2, \ldots, K$ do
    The available client set changes in the current $k$-th time zone.
    The server senses the change of available client set and stars following model branch switching.
    Maintain the shared feature extractor layers $\boldsymbol{\theta}_f$.
    Switch the specialized decision head to corresponding $\boldsymbol{\theta}_k$ for current client set $k$.
    Form the new DRL strategy network $\boldsymbol{\theta} = (\boldsymbol{\theta}_f, \boldsymbol{\theta}_k)$.
    Conduct DRL training to update network parameter $\boldsymbol{\theta} = (\boldsymbol{\theta}_f, \boldsymbol{\theta}_k)$ until next time zone change.
end for
Repeat the DRL training on multi-branch strategy network until all model branches reach convergence.
```

B. Training & Inference Workflow

Different from the traditional DRL training on single model, we optimize the multi-branch network by an alternative manner, which is summarized in Algorithm 1. First, we randomly initialize all parameters of the multi-branch strategy network

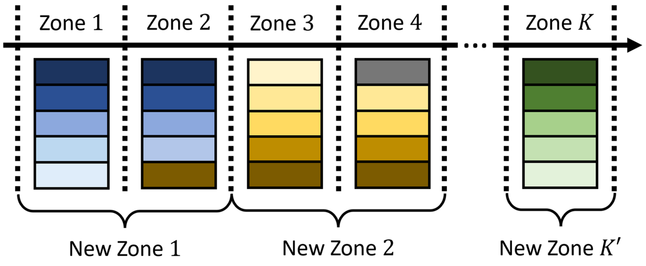


Fig. 5. Demonstration of neighborhood clustering based on client distribution similarity. The black bold line indicates the timeline and the dashed lines indicate 24-hour division in real-world. Different colors in each time slots denote different clients.

(Line 1). Then, the available client set will change in the current k-th time zone, where some old clients exit and other new clients join in (Line 2). Next, the server senses the dynamic change of the available client set and conducts the model branch switching to choose the specialized branch for client set k in k-th time zone (Line 4 to 7). The subsequent training will be conducted on the new model branch until the dynamic change is sensed by server again and repeat the previous process. Finally, the DRL training of multi-branch strategy network will keep going until all model branches reach convergence (Line 8).

The inference workflow of our multi-branch strategy network is based on the sensing of available client set change on the timeline. In this paper, we assume that the available time zones for each client are stable, which means that each client will only connect with the server in their own pre-defined available time zones. For easy understanding, we also set the change of available client set only to occur at every whole time (e.g., 10:00).

C. Automatic Time Zone Division

Client Time Zone Distribution. We assume that each client has total 8 available time slots with 1-hour each, which means they will connect with the server for 8 hours in a day. Besides, we assume the number of connected clients in each 1-hour time slot is the same on server-side. Therefore, under a real-world 24-hour time zone division, in each 1-hour time slot, the server will connect with $\frac{8\tilde{}N}{24} = \frac{N}{3}$ clients simultaneously. As shown in Fig. 5, if we set total $N = 15$ clients in our FL system, the server will connect with $\frac{8*15}{24} = 5$ clients in each 1-hour time slot, where different clients are indicated by different colors. At every whole time (dashed line), some old clients will exit and other new clients will join, which makes the number of connected clients unchanged.

Automatic Time Zone Division by Neighborhood Clustering. Although the previous fixed 24-hour time zone division is enough to cover many real-world FL applications, we can observe that the available client set is slowly changing and the majority is unchanged. For example, clients will always connect with the server in the daytime and exit the FL system in the nighttime, which means the majority of connected clients will maintain unchanged for several time slots. This phenomenon indicates that we don't need to build separated model branches for two neighborhood time slots, where their available client sets are very similar.

Algorithm 2: The Neighborhood Clustering Based on Client Distribution Similarity.

```
Input: Client distribution $\mathcal{C}$, Similarity threshold $\tau$.
Output: Optimal time zone division.
 1: for time zone $k = 1, 2, \ldots, K$ do
 2:   $s_{k-1} = \text{similarity}(\mathcal{C}_k, \mathcal{C}_{k-1})$.
 3:   $s_{k+1} = \text{similarity}(\mathcal{C}_k, \mathcal{C}_{k+1})$.
 4:   if $s_{k-1} \geq s_{k+1}$ and $s_{k-1} \geq \tau$ then
 5:     Merge time zone $k$ and $k-1$.
 6:   else if $s_{k+1} \geq s_{k-1}$ and $s_{k+1} \geq \tau$ then
 7:     Merge time zone $k$ and $k+1$.
 8:   else
 9:     Time zone $k$ remains unchanged.
10:   end if
11: end for
12: update the new time zone list $[1, \ldots, K]$ and repeat
      above process until convergence
```

Therefore, to further reduce the training cost of the multi-branch network, we propose an automatic time zone division by neighborhood clustering. The key insights of neighborhood clustering are based on the client distribution similarity between adjacent time zones, if the similarity is higher than a pre-defined threshold $\tau \in [0, 1]$, we will merge these two adjacent time slots into the same time zone. Finally, we can generate a new time zone division from the previous 24-hour division. The details of neighborhood clustering are illustrated in Algorithm 2

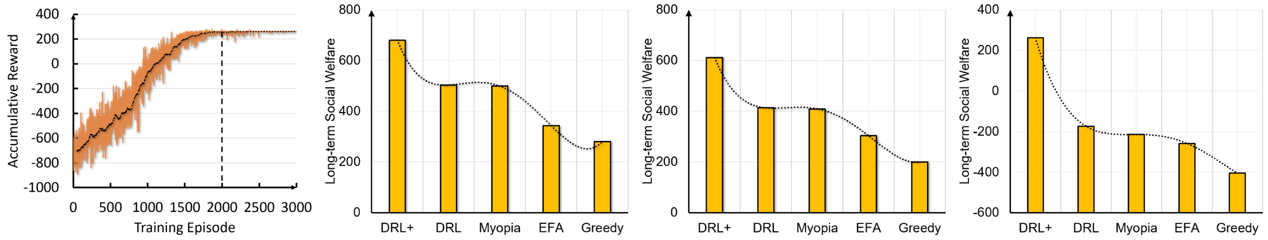
First, the server will obtain the client time zone distribution based on historical experience estimation and define a similarity threshold $\tau \in [0, 1]$ (normally a large number such as $\tau = 0.9$). Then, each time zone will compute its client distribution similarity with its neighborhoods (Line 2 to 3). If the client distribution similarity reaches the threshold requirements, merging itself with its adjacent neighbors, else remaining unchanged (Line 4 to 10). This process will repeat until convergence to generate a new time zone division.

VII. Performance Evaluation

A. Experiment Setup

Our FL system framework is based on Google's *FedAvg*, where all experiments are conducted on three real-world datasets including MNIST, Fashion-MNIST, and CIFAR-10. The FL model parameters and structures for image tasks on different datasets exactly follow the setting in [1].

Federated Learning System Setup: The system setup on different datasets is the same. For client data Non-IID distribution, we randomly allocate the entire dataset to all clients in the FL system, where each data sample is uniquely owned by one client (Non-overlapping). Different Non-IID degrees of data distribution is controlled by setting different random seeds. The Mini-batch Stochastic Gradient Descent (SGD) is adopted as the optimizer for the local FL model training on the client-side with a mini-batch size of 20 and a learning rate of 0.01. For the parameters related to system energy cost modeling, the number of CPU cycles required to execute one sample c is 20



(a) The reward convergence of multi-branch DRL network. (b) Long-term social welfare comparison on MNIST dataset. (c) Long-term social welfare comparison on Fashion-MNIST dataset. (d) Long-term social welfare comparison on CIFAR-10 dataset.

Fig. 6. Performance of our DRL-based long-term mechanism and its comparison with other benchmarks. **"DRL+"** indicates our new multi-branch network, while **"DRL"** indicates the old single network. The dashed line in Fig. 6(a) indicates the training convergence point of DRL strategy network, while other dashed lines in Fig. 6(b)-(d) shows the change tendency.

cycles/bit. To simulate client resource heterogeneity in the real world, we set the effective capacitance coefficient β_i of different clients' computation chip-set randomly distributed within $1 \sim 2 \times 10^{-28}$, and the CPU-cycle frequency δ_i is randomly distributed within $1 \sim 2$ GHz. The preference adjustment coefficient of server defaults to $\lambda = 1000$ on the trade-off between accuracy improvement and system training cost. The influence of different λ values on the trade-off is also investigated by various experiments in Fig. 8.

Deep Reinforcement Learning Mechanism Setup: We adopt the PPO algorithm as the basis of our DRL method [51], which is an advanced version of actor-critic network. To provide enough historical experience for the training of DRL strategy network, we set the total training episode number $E = 3000$, where the step number in each episode is 5. The learning rate of actor and critic network in PPO is set as 0.00003, which will decay by 95% every 40 episodes. To drive the DRL agent to achieve our long-term optimization objective, we stop the intermediate update until the whole episode is finished, and update the strategy of DRL agent by the long-term accumulative reward $R = \sum_{t=0}^{T} \gamma^t r_t$ at the end, where the reward discount factor is $\gamma = 0.95$.

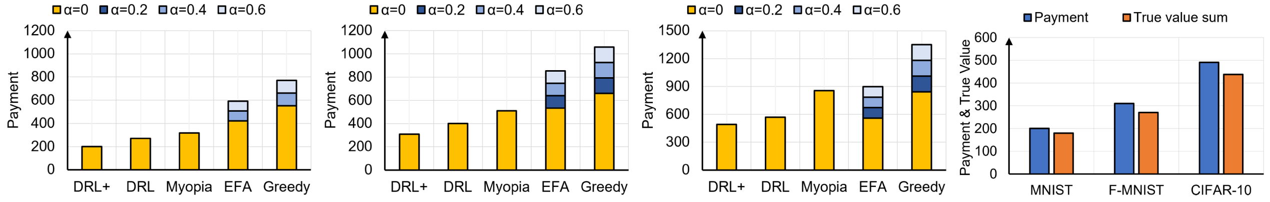
Multi-Branch DRL Strategy Network Setup: The multi-branch DRL strategy network consists of two parts: 1) The common feature extractor $\boldsymbol{\theta}_f$, which are set as two-layer MLP networks, and each layer has 512 neurons with ReLu activation. 2) The following specialized decision heads $\boldsymbol{\theta}_k$, which is defined as a N-dimensional fully-connected layer with a final softmax output, and N is the client number. Fig. 4 illustrates the whole structure of our multi-branch DRL network, which is based on the assumption of stable periodically client shifting. The details of client time zone distribution settings are elaborated in Section VI-C.

Comparison Benchmarks: We compare our method with several other benchmark approaches as illustrated below: *1) Myopia*: The entire long-term optimization problem will be decomposed into a series of independent one-round sub-problems, which are widely accepted by many online auction solutions in other studies [10], [11], [12]. Each sub-problem is optimally solved with an exhaustive search in its solution space and the payment rule strictly follows VCG. Obviously, this decomposition will break the continual correlation between adjacent FL rounds, which results in a sub-optimal solution from the long-term perspective. *2) Expert-guided FedAvg (EFA)*: This benchmark is based on the traditional FedAvg algorithms, where the local epoch number is manually set according to the historical training experience. *3) ϵ-Greedy*: This benchmark is a heuristic algorithm that always chooses the strategy with maximal reward from the training experience replay buffer. To achieve the trade-off between exploring and exploiting, we set a probability of 80% to follow above greedy strategy, while the remaining 20% probability will generate a random strategy.

B. Performance of Long-Term Incentive Mechanism

In Fig. 6, we use various experiment results to show the performance of our incentive mechanism on long-term social welfare optimization. First, Fig. 6(a) shows the reward convergence tendency of the multi-branch DRL strategy network in the training process. The accumulative reward (i.e., the FL system long-term social welfare) keeps growing during the training process and reaches the convergence point at around 2000-th episode, which indicates that our DRL agent has learned the optimal strategy ψ^* for the long-term optimization.

Then, extensive experiments on different datasets (MNIST, Fashion-MNIST and CIFAR-10) further demonstrate the superiority of our multi-branch DRL-based incentive mechanism ("DRL+" in Fig. 6) by comparison with other benchmarks. We can observe obvious gaps between them in the achieved long-term social welfare. Taking the results on MINST in Fig. 6(b) as an example, the old single DRL incentive mechanism of the conference version ("DRL" in Fig. 6) cannot adapt to the periodically client shifting environment, which results in a relatively low long-term social welfare optimization. The *Myopia* can achieve a relatively high long-term social welfare since its strategy is optimal for each independent single-round sub-problem. However, the problem decomposition of *Myopia* makes it unable to sense the continual correlation between adjacent rounds, thus failing in long-term optimization compared to our method (36% gap). The *EFA* can outperform *Greedy* since its strategy is manually adjusted according to our historical training experience, which is much more efficient than the heuristic *Greedy* with randomness. The performance situations on the Fashion-MNIST dataset in Fig. 6(c) are very similar to MNIST due to their similar dataset features. Besides, the experiment results on CIFAR-10 in Fig. 6(d) show a higher performance gap of our DRL-based method between other benchmarks, since the



(a) The payment comparison on MNIST dataset. (b) The payment comparison on F-MNIST dataset. (c) The payment comparison on CIFAR-10 dataset. (d) The comparison between payment and client true value.

Fig. 7. Payment comparison on different benchmarks and datasets are shown in Fig. 7(a) to (c), where α is the parameter to simulate different untruthfulness levels. **"DRL+"** indicates our new multi-branch network, while **"DRL"** indicates the old single network. Fig. 7(d) is experiment proof of the client's long-term individual rationality (IR) property in Eq. (13) on all datasets.

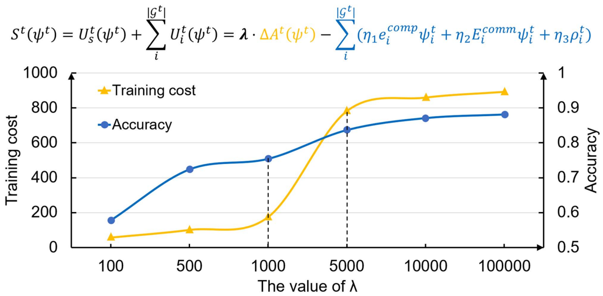


Fig. 8. Change trend between accuracy and training cost under different values of λ on MNIST, where λ is a preference adjustment parameter. The trade-off equation is attached for reference.

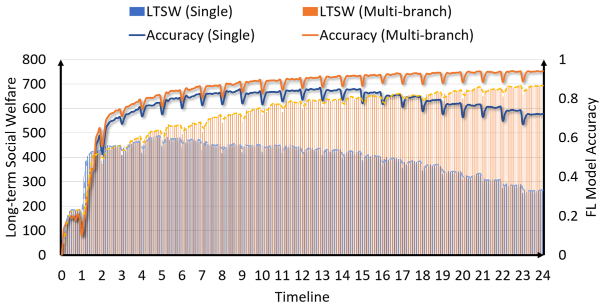


Fig. 9. Long-term Social Welfare (LTSW) & Accuracy tendency under client periodically shifting with Single vs. Multi-branch Network. We uses a dual coordinate system (LTSW on the left and Accuracy on the right). The bold lines indicated the Accuracy and the shadow bar areas indicate LTSW.

task complexity of CIFAR-10 is much higher than other datasets. It's more difficult for these non-learning-based benchmarks to learn a good strategy, while our method can still learn an optimal long-term strategy from complex environments.

Furthermore, recalling the definition of λ in (8) and (9), which is the preference adjustment parameter for the trade-off between the accuracy improvement and training cost. A higher λ value means the server will concern more about the accuracy improvement and ignore the training cost, and vice versa. We conduct various experiments to show the influence of different λ values on this trade-off in Fig. 8. As the increasing of λ value, the model accuracy and training cost are both growing with different rates, where two key turning points ($\lambda = 1000$ and 5000) are marked by dashed lines. When $100 < \lambda < 1000$, the strategy will transform from the extreme strategy (choosing the most cost-saving clients) to the optimal strategy (choosing diverse client combinations to improve accuracy while saving cost). When $1000 < \lambda < 5000$, the comparable proportion of accuracy and cost make our mechanism sensitive to their trade-off, the improvement of accuracy will consume much more cost than before. When $\lambda > 5000$, the strategy is transformed to another extreme that only improves the accuracy and ignores the cost.

Fig. 7 shows the payment comparison of all methods. Since *EFA* and *Greedy* cannot ensure truthfulness, clients can scam more payments from the server by lying about their costs. For a clear comparison, we add a parameter α to simulate the payments under different untruthfulness levels through multiplying the true cost by $(1 + \alpha)$. In particular, the $\alpha = 0$ for our method and *Myopia* since they can ensure truthfulness. From Fig. 7(a), (b), and (c), we can observe that minimal truthful payments are achieved by our DRL-based method (reduced by 37% on MNIST, 39% on Fashion-MNIST, and60% on CIFAR-10), which is simultaneously under an optimal long-term social welfare guarantee. Besides, to further prove the long-term individual rationality property of our DRL-based method in (8) and (9), we compare the received payments and the true value (true training cost) sum of all clients in the whole FL process in Fig. 7(d), where the results clearly indicate that the individual rationality property is satisfied.

C. Performance of Multi-Branch Network

In this section, we show the performance of the traditional single network and our multi-branch network, facing the client periodically shifting problem in Fig. 9. The traditional single model fails to handle this problem due to its structural limitations, which leads to the droppings in both accuracy and long-term social welfare as the change of timeline. In contrast, our multi-branch network design is sensitive to the available client set change and can adaptively switch to corresponding decision heads to handle time zone changes, which results in overall smooth growth. Besides, we also illustrate the advantage of neighborhood clustering compared with the naive 24-hour division in Table II. Compared with single network, the 24-branch network can significantly improve long-term social welfare by 140% with a 33% training time increase. After using neighborhood clustering, the performance of long-term social welfare can maintain the same while the training time only increases by 12.9%.

TABLE II
PERFORMANCE OF NEIGHBORHOOD CLUSTERING ON MULTI-BRANCH NETWORK

Performance Comparison	Branch Number	Training Time ($\times 10^4$/s)	Long-term Social Welfare
Single	1	12.96	285.73
Multi-branch (24-hour)	24	17.25	686.57
Multi-branch (clustering)	10	**14.64**	**681.92**

VIII. CONCLUSION

This paper focuses on designing an adaptive long-term incentive mechanism for the sustainable FL system with a practical periodical client shifting problem. First, considering the long-term optimization objectives, we present an online VCG auction modeling for the FL system to guarantee critical economic properties for system sustainability. Second, we adopt a deep reinforcement learning-based method to achieve long-term objectives optimization. Third, we design a multi-branch DRL strategy network to replace the original single network, which can adaptively switch different decision heads as the change of available client sets on the timeline. Compared with state-of-the-art approaches, the long-term social welfare of FL system increases by 36% with a 37% reduction in payment. Besides, the multi-branch network can adaptively handle periodical client shifting.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Artif. Intell. Statist.*, PMLR, 2017, pp. 1273–1282.

[2] A. Hard et al., “Federated learning for mobile keyboard prediction,” 2018, *arXiv: 1811.03604*.

[3] Y. Zhan and J. Zhang, “An incentive mechanism design for efficient edge learning by deep reinforcement learning approach,” in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 2489–2498.

[4] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, “A learning-based incentive mechanism for federated learning,” *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[5] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, “Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 700–10 714, Dec. 2019.

[6] W. Hediger, “Sustainable development and social welfare,” *Ecological Econ.*, vol. 32, no. 3, pp. 481–492, 2000.

[7] J. Midgley, *Social Development: The Developmental Perspective in Social Welfare*. Newbury Park, CA, USA: Sage, 1995.

[8] G. B. Asheim, T. Mitra, and B. Tungodden, “Sustainable recursive social welfare functions,” in *The Economics of the Global Environment*. Berlin, Germany: Springer, 2016, pp. 165–190.

[9] R. Zhou, J. Pang, Z. Wang, J. C. Lui, and Z. Li, “A truthful procurement auction for incentivizing heterogeneous clients in federated learning,” in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst.*, 2021, pp. 183–193.

[10] W. Shi, L. Zhang, C. Wu, Z. Li, and F. C. Lau, “An online auction framework for dynamic resource provisioning in cloud computing,” *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 71–83, Jun. 2014. [Online]. Available: https://doi.org/10.1145/2637364.2591980

[11] S. Chen, L. Jiao, L. Wang, and F. Liu, “An online market mechanism for edge emergency demand response via cloudlet control,” in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 2566–2574.

[12] W. You, L. Jiao, J. Li, and R. Zhou, “Scheduling DDoS cloud scrubbing in ISP networks via randomized online auctions,” in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1658–1667.

[13] H. Eichner, T. Koren, B. McMahan, N. Srebro, and K. Talwar, “Semi-cyclic stochastic gradient descent,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 1764–1773.

[14] Y. Ding et al., “Distributed optimization over block-cyclic data,” 2020, *arXiv: 2002.07454*.

[15] V. Krishna, *Auction Theory*, 2nd ed. New York, NY, USA: Academic Press, 2010.

[16] W. Vickrey, “Counterspeculation, auctions, and competitive sealed tenders,” *J. Finance*, vol. 16, no. 1, pp. 8–37, 1961. [Online]. Available: http://www.jstor.org/stable/2977633

[17] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[18] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, “A survey of incentive mechanism design for federated learning,” *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 2, pp. 1035–1044, Second Quarter 2022.

[19] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, “DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive,” *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2438–2455, Sep./Oct. 2021.

[20] U. Majeed and C. S. Hong, “FLchain: Federated learning via MEC-enabled blockchain network,” in *Proc. 20th Asia-Pacific Netw. Operations Manage. Symp.*, 2019, pp. 1–4.

[21] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, “A crowdsourcing framework for on-device federated learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3241–3256, May 2020.

[22] L. U. Khan et al., “Federated learning for edge networks: Resource optimization and incentive mechanism,” *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 88–93, Oct. 2020.

[23] Y. Liu, L. Wu, Y. Zhan, S. Guo, and Z. Hong, “Incentive-driven long-term optimization for edge learning by hierarchical reinforcement mechanism,” in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst.*, 2021, pp. 35–45.

[24] R. Zeng, S. Zhang, J. Wang, and X. Chu, “FMore: An incentive scheme of multi-dimensional auction for federated learning in MEC,” in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 278–288.

[25] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, “Toward an automated auction framework for wireless federated learning services market,” *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3034–3048, Oct. 2021.

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

[27] Y. Deng et al., “FAIR: Quality-aware federated learning with precise user incentive and model aggregation,” in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[28] M. Tang and V. W. Wong, “An incentive mechanism for cross-silo federated learning: A public goods perspective,” in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[29] J. Xu and H. Wang, “Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2021.

[30] P. Dütting, Z. Feng, H. Narasimhan, D. Parkes, and S. S. Ravindranath, “Optimal auctions through deep learning,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 1706–1715.

[31] P. Dütting, Z. Feng, H. Narasimhan, D. C. Parkes, and S. S. Ravindranath, “Optimal auctions through deep learning,” *Commun. ACM*, vol. 64, no. 8, pp. 109–116, Jul. 2021. [Online]. Available: https://doi-org.ezproxy.lb.polyu.edu.hk/10.1145/3470442

[32] J. Rahme, S. Jelassi, and S. M. Weinberg, “Auction learning as a two-player game,” in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YHdeAO61l6T

[33] Z. Feng, H. Narasimhan, and D. C. Parkes, “Deep learning for revenue-optimal auctions with budgets,” in *Proc. 17th Int. Conf. Auton. Agents MultiAgent Syst.*, Richland, SC, 2018, pp. 354–362.

[34] N. Golowich, H. Narasimhan, and D. C. Parkes, “Deep learning for multi-facility location mechanism design,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 261–267. [Online]. Available: https://doi.org/10.24963/ijcai.2018/36

[35] N. C. Luong, Z. Xiong, P. Wang, and D. Niyato, “Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach,” in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.

[36] X. Liu et al., “Neural auction: End-to-end learning of auction mechanisms for e-commerce advertising,” in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2021, pp. 3354–3364. [Online]. Available: https://doi.org/10.1145/3447548.3467103

[37] P. Tang, “Reinforcement mechanism design,” in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 5146–5150. [Online]. Available: https://doi.org/10.24963/ijcai.2017/739

[38] Q. Cai, A. Filos-Ratsikas, P. Tang, and Y. Zhang, “Reinforcement mechanism design for e-commerce,” in *Proc. World Wide Web Conf.*, Republic and Canton of Geneva, CHE, 2018, pp. 1339–1348. [Online]. Available: https://doi.org/10.1145/3178876.3186039

[39] Z. Zhang et al., “Optimizing multiple performance metrics with deep GSP auctions for e-commerce advertising,” in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, New York, NY, USA, 2021, pp. 993–1001. [Online]. Available: https://doi.org/10.1145/3437963.3441771

[40] H. Yu et al., “A fairness-aware incentive scheme for federated learning,” in *Proc. AAAI/ACM Conf. AI Ethics Soc.*, 2020, pp. 393–399.

[41] T. D. Burd and R. W. Brodersen, “Processor design for portable systems,” *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 13, no. 2/3, pp. 203–221, 1996.

[42] A. Ghorbani and J. Zou, “Data Shapley: Equitable valuation of data for machine learning,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 2242–2251.

[43] E. Breck, N. Polyzotis, S. Roy, S. Whang, and M. Zinkevich, “Data validation for machine learning,” in *Proc. Mach. Learn. Syst.*, 2019, pp. 334–347.

[44] S. Tang et al., “Data valuation for medical imaging using Shapley value and application to a large-scale chest X-ray dataset,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–9, 2021.

[45] M. L. Fisher, J. H. Hammond, W. R. Obermeyer, and A. Raman, “Making supply meet demand in an uncertain world,” *Harvard Bus. Rev.*, vol. 72, pp. 83–83, 1994.

[46] J.-P. Bouchaud, J. D. Farmer, and F. Lillo, “How markets slowly digest changes in supply and demand,” in *Handbook of Financial Markets: Dynamics and Evolution*. Amsterdam, The Netherlands: Elsevier, 2009, pp. 57–160.

[47] M. de Oliveira Antunes and F. P. de Almeida Prado, “A housing price dynamics model using heterogeneous interacting agents,” *SIAM J. Appl. Math.*, vol. 78, no. 5, pp. 2648–2671, 2018. [Online]. Available: https://doi.org/10.1137/17M1145215

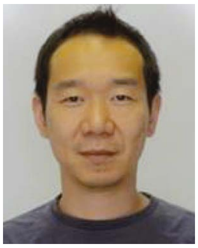
[48] H. Tian, M. Yu, and W. Wang, “Continuum: A platform for cost-aware, low-latency continual learning,” in *Proc. ACM Symp. Cloud Comput.*, 2018, pp. 26–40.

[49] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[50] R. Caruana, “Multitask learning,” *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[51] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017, *arXiv: 1707.06347*.

Leijie Wu (Student Member, IEEE) received the BEng degree of science in automation from the School of Xuteli, Beijing Institute of Technology, Beijing, China, in 2019. He is currently working toward the PhD degree with the Department of Computing in The Hong Kong Polytechnic University, Hong Kong, China. His current research interest includes federated learning, mobile edge computing, deep reinforcement learning, and incentive mechanism design.



Song Guo (Fellow, IEEE) is a full professor with the Department of Computing, The Hong Kong Polytechnic University. He also holds a Changjiang chair professorship awarded by the Ministry of Education of China. He is a fellow of the Canadian Academy of Engineering. His research interests are mainly in edge AI, machine learning, mobile computing, and distributed systems. He published many papers in top venues with wide impact in these areas and was recognized as a Highly Cited researcher (Clarivate Web of Science). He is the recipient of more than a dozen Best Paper Awards from IEEE/ACM conferences, journals, and technical committees. He is the editor-in-chief of IEEE Open Journal of the Computer Society and the chair of IEEE Communications Society (ComSoc) Space and Satellite Communications Technical Committee. He was an IEEE ComSoc distinguished lecturer and a member of IEEE ComSoc Board of Governors. He has served for IEEE Computer Society on fellow Evaluation Committee, and been named on editorial board of a number of prestigious international journals like *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Emerging Topics in Computing*, etc. He has also served as chairs of organizing and technical committees of many international conferences.

Zicong Hong (Graduate Student Member, IEEE) received the BEng degree in software engineering from the School of Data and Computer Science, Sun Yat-sen University. He is currently working toward the PhD degree with the Department of Computing in The Hong Kong Polytechnic University. His current research interest includes blockchain, edge/cloud computing, and federated learning.

Yi Liu (Student Member, IEEE) received the BEng degree in automation from the School of Astronautics, Harbin Institute of Technology, in 2018. He is currently working toward the PhD degree with the Department of Computing in The Hong Kong Polytechnic University. His current research interest include mobile edge learning, deep reinforcement learning, model pruning, out-of-distribution generalization, and self-supervised learning.

Wenchao Xu (Member, IEEE) received the BE and ME degrees from Zhejiang University, Hangzhou, China, in 2008 and 2011, respectively, and the PhD degree from the University of Waterloo, Canada, in 2018. He is currently a research assistant professor with The Hong Kong Polytechnic University. He has been an assistant professor with the School of Computing and Information Sciences in Caritas Institute of Higher Education, Hong Kong, and a software engineer with Alcatel Lucent Shanghai Bell Company Ltd., Shanghai. His research interests include distributed computing, Internet of Vehicles, and wireless communication.

Yufeng Zhan received the PhD degree from the Beijing Institute of Technology (BIT), Beijing, China, in 2018. He is currently an assistant professor with the School of Automation with BIT. Prior to join BIT, he was a post-doctoral fellow with the Department of Computing with The Hong Kong Polytechnic University. His research interests include Internet of Things, cloud/edge computing, and machine learning system.