Pareto Actor-Critic for Communication and Computation Co-Optimization in Non-Cooperative Federated Learning Services

Renxuan Tan, *Student Member, IEEE*, Rongpeng Li, *Senior Member, IEEE*, Xiaoxue Yu, *Student Member, IEEE*, Xianfu Chen, *Senior Member, IEEE*, Xing Xu, and Zhifeng Zhao, *Senior Member, IEEE*

***Abstract*—Federated learning (FL) in multi-service provider (SP) ecosystems is fundamentally hampered by non-cooperative dynamics, where privacy constraints and competing interests preclude the centralized optimization of multi-SP communication and computation resources. In this paper, we introduce `PAC-MCoFL`, a game-theoretic multi-agent reinforcement learning (MARL) framework where SPs act as agents to jointly optimize client assignment, adaptive quantization, and resource allocation. Within the framework, we integrate Pareto Actor-Critic (PAC) principles with expectile regression, enabling agents to conjecture optimal joint policies to achieve Pareto-optimal equilibria while modeling heterogeneous risk profiles. To manage the high-dimensional action space, we devise a ternary Cartesian decomposition (TCAD) mechanism that facilitates fine-grained control. Further, we develop `PAC-MCoFL-p`, a scalable variant featuring a parameterized conjecture generator that substantially reduces computational complexity with a provably bounded error. Alongside theoretical convergence guarantees, our framework's superiority is validated through extensive simulations – `PAC-MCoFL` achieves approximately 5.8% and 4.2% improvements in total reward and hypervolume indicator (HVI), respectively, over the latest MARL solutions. The results also demonstrate that our method can more effectively balance individual SP and system performance in scaled deployments and under diverse data heterogeneity.**

***Index Terms*—Federated learning, multi-agent reinforcement learning, Pareto actor-critic, communication and computation co-optimization.**

Received 22 April 2025; revised 14 July 2025; accepted 19 August 2025. Date of publication 22 August 2025; date of current version 9 January 2026. This work was supported in part by the National Key Research and Development Program of China under Grant 2024YFE0200600, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LR23F010005, in part by Huawei Cooperation Project under Grant TC20240829036, and in part by the Big Data and Intelligent Computing Key Lab of CQUPT under Grant BDIC-2023-B-001. The work of Xing Xu was supported by the Hebei Natural Science Foundation of China under Grant F2025521001. The work of Xianfu Chen was supported by the National Natural Science Foundation of China under Grant U24A20209. Recommended for acceptance by S. Mazuelas. *(Corresponding author: Rongpeng Li.)*

Renxuan Tan, Rongpeng Li, and Xiaoxue Yu are with the College of Information Science and Electronic Engineering, Zhejiang University, Zhejiang 310027, China (e-mail: ttrx@zju.edu.cn; lirongpeng@zju.edu.cn; sdwhyxx@zju.edu.cn).

Xianfu Chen is with Shenzhen CyberAray Network Technology Company Ltd., Shenzhen 518038, China (e-mail: xianfu.chen@ieee.org).

Xing Xu is with the Information and Communication Branch of State Grid Hebei Electric Power Company Ltd., Shenzhen 518038, China (e-mail: hsuxing@zju.edu.cn).

Zhifeng Zhao is with Zhejiang Lab, Zhejiang University, Zhejiang 310027, China (e-mail: zhaozf@zhejianglab.com).

Digital Object Identifier 10.1109/TMC.2025.3601833

I. Introduction

With the rapid proliferation of smart devices, federated learning (FL) has emerged due to its privacy-friendly nature to facilitate distributed learning. Beyond the well-studied scenario of a single service provider (SP) orchestrating FL [1], the burgeoning complexity of client-side applications necessitates a multi-provider ecosystem. Real-world scenarios include Apple's private cloud compute (PCC) [2], where multiple third-party services (via PCC encrypted proxies) interact with a shared pool of end-user devices. Compared with a single SP, this multi-SP setting, where SPs share finite communication and computational resources [3], introduces significant new challenges. Particularly, due to stringent privacy and competitiveness imperatives, there naturally emerges an inherent reluctance among SPs to share model details and co-optimization strategies. This reluctance becomes more pronounced for sharing the quality of service (QoS) metrics and operational trade-offs, precluding fully cooperative optimization. Under such partially observable environment, where only limited metadata on resource coordination is shared via third-party, regulatory-compliant channels [4], there emerges a strong incentive to develop non-cooperative solutions.

Typically, the communication and computation co-optimization in FL shall simultaneously take into account client selection, resource allocation, and the dynamically changing network environment [5]. The high dimensionality of the decision space, coupled with complex inter-variable dependencies, renders this multi-objective problem intractable for conventional optimization solvers [6]. In this regard, deep reinforcement learning (DRL), which learns optimal policies without prior statistical knowledge, offers a promising avenue to devise adaptive strategies. Recent advances in DRL-based optimization have demonstrated its potential in various facets of FL, including client selection [7], hyperparameter adaptive tuning [8], bandwidth partitioning [9], power management [10], and energy consumption scheduling [11]. Nevertheless, these studies predominantly assume a single SP and cannot be straightforwardly extended to a multi-SP scenario, due to the non-cooperative nature [12].

Multi-agent reinforcement learning (MARL) offers a promising framework to address the intricate co-optimization problem

therein [13], [14], [15]. However, classical MARL algorithms overlook the impact of joint actions and often promote individual actions lacking synergy in early training stages [16]. Furthermore, value function estimation under the mean squared error criterion implicitly assumes symmetric risk preferences [17], and fails to capture the asymmetric risk attitudes commonly exhibited by SPs (e.g. conservative versus aggressive policy). Consequently, games in MARL may converge to a risk-averse equilibrium rather than a Pareto-optimal one [18], yielding suboptimal resource allocation, degraded service quality, and inefficient global resource utilization. The Pareto actor-critic (PAC) framework [19] lays the groundwork for addressing equilibrium selection in MARL systems, yet its practical application to multi-SP FL is severely limited by its reliance on computationally prohibitive brute-force policy conjectures and restrictive single-dimensional action control.

In this paper, we propose a PAC-based communication and computation co-optimization scheme `PAC-MCoFL` for non-cooperative multi-SP FL services. `PAC-MCoFL` treats each FL SP as a PAC agent, which autonomously determines its operational policy and resource allocation (encompassing quantization levels, aggregation strategies, computational resource assignments, and bandwidth allocation) based on environmental observations and inferred actions of other agents. Unlike classical PAC implementations [19], we first develop a critic network grounded in expectile regression to capture heterogeneous risk preferences among SPs, enabling a more granular and realistic characterization of SPs. Meanwhile, a ternary Cartesian action decomposition (TCAD) mechanism is introduced to facilitate fine-grained, multi-dimensional control over SP operations. Furthermore, we introduce `PAC-MCoFL-p`, a variant of `PAC-MCoFL`, by replacing the computationally intensive brute-force conjecture of joint actions with a neural generator. This generator can be seamlessly integrated into `PAC-MCoFL` through end-to-end training, achieving comparable performance with provably bounded approximate error. In brief, the main contributions of this research can be summarized as follows.

- We propose `PAC-MCoFL`, a framework that enables decentralized communication and computation co-optimization in partially observable, non-cooperative multi-SP FL systems. `PAC-MCoFL` adaptively optimizes operational policies and resource allocation, leveraging game-theoretic principles to conjecture actions from other agents, thereby mitigating convergence to suboptimal equilibria. It incorporates two key mechanisms: (i) an expectile regression-based critic for capturing asymmetric risk profiles among SPs, and (ii) a TCAD mechanism for fine-grained multi-dimensional action space navigation.
- We develop `PAC-MCoFL-p` to overcome the scalability bottleneck of exhaustive action conjecture in `PAC-MCoFL`. This practical variant features a parameterized conjecture generator that substantially reduces computational complexity, making the solution feasible for large-scale multi-SP scenarios. We formally establish that this approximation has a provably bounded error, ensuring its reliability.
- We provide a theoretical analysis of `PAC-MCoFL`, focusing on the convergence property of Q-function. Extensive simulations have validated the superiority and robustness of our proposed algorithm.

The remainder of this paper is organized as follows. Section II summarizes relevant literature. Section III introduces the system model and formulates the problem. Section IV presents the `PAC-MCoFL` algorithm, while Section V provides simulation results and analyses to highlight the significance of our method. Finally, Section VI concludes the paper.

II. Related Works

Efficient FL with Single SP: Improving communication and computation efficiency in FL has received significant attention. Approaches like AdaQuantFL [24] and FedDQ [25] dynamically adjust quantization levels to balance communication cost and convergence. FedDrop [26] applies heterogeneous dropout to prune global models into personalized subnets, reducing both communication and computation loads. Moreover, FedEco [20] employs adaptive hyperparameter tuning to better exploit client-side computing resources, significantly lowering energy consumption across participating devices. However, these static or heuristic-based methods often fail to adapt to the dynamic and heterogeneous nature of real-world FL environments. To overcome this, recent works have applied DRL to resource optimization in wireless systems like mobile edge computing (MEC) and internet of vehicles (IoV) [27], [28]. For example, [29] uses a soft actor-critic (SAC) framework for context-aware power control in IoV-MEC scenarios, while [30] adopts a double deep Q-network (DDQN)-based approach to jointly optimize energy, latency, and communication overheads. Within the FL domain, [9] employs REINFORCE to automate client selection and bandwidth allocation strategies based on historical feedback, reducing latency and energy while enhancing long-term FL performance. Similarly, a proximal policy optimization (PPO)-based scheme in [8] dynamically adjusts the quantization interval during global model distribution, accelerating convergence and improving accuracy.

Efficient FL with multi-SP: FL involving multiple SPs presents additional complexity due to the heterogeneous configurations, competing interests, and the inherent asymmetry risks of SPs [12], [22], [23]. These challenges necessitate adaptive and decentralized coordination schemes for efficient FL across SPs. To address client heterogeneity, [23] proposes a multi-core FL architecture where multiple global models — ranging from shallow to deep — are concurrently trained, with task inter-dependencies explicitly modeled. Resource allocation strategies among SPs have also been explored. Centralized [22] and distributed [12] optimization methods are proposed to allocate bandwidth and CPU power. However, most of these approaches overlook practical power constraints. Under limited power budgets, [21] frames the joint optimization of client selection and bandwidth allocation as a novel variant of the knapsack problem [31] to accelerate multiple FL models' training in wireless networks. Game-theoretic models have also been applied to capture the strategic behavior of SPs. Ref. [32] designs a multi-leader-follower game for multi-task FL, establishing equilibrium between SPs and clients. A two-stage Stackelberg

TABLE I
THE SUMMARY OF DIFFERENCES WITH RELATED LITERATURE

References	multi-SP	Client Assignment	Adaptive Quantization	Computation Efficiency	Communication Efficiency	Brief Description
[7], [9]	○	●	○	○	●	Over-ideal assumption that heterogeneous services are scheduled by the same server.
[8]	○	●	○	●	●	A simple single-agent model.
[20]	○	○	●	●	○	Only energy efficiency analysis with traditional convex optimization methods.
[12], [21]	●	○	○	○	●	Absence of analysis of energy consumption.
[22], [23]	●	○	○	●	●	Neglection of fine-grained optimization within a service.
This work	●	●	●	●	●	For a FL with multiple SPs, a partial observable, non-cooperative game, involving *expectile regression*, *TCAD* and *parameterized conjecture generator*, is introduced to achieve joint optimization of communication and computation.

Notations: ○ *indicates not included*; ● *indicates fully included.*

game in [33] co-optimizes client selection and resource allocation, improving cluster efficiency while reducing FL costs. Despite these advancements, the majority of existing studies have largely downplayed the significance of non-cooperative game dynamics among multiple SPs.

Equilibrium Games: Nash equilibrium (NE) represents a milestone in multi-agent stochastic games [34]. Over decades, MARL has emerged as a powerful approach for dynamically identifying NE equilibria in complex, multi-agent settings, effectively bridging classical game theory with modern computational techniques [35], [36], [37]. In friend-and-foe Q-learning [35], an agent r assumes that its "friend agent", $-r$, voluntarily selects actions that maximize r's joint Q-value. Conversely, foe Q-learning is used in adversarial settings [35], where each agent seeks to minimize the opponent's expected return. Ref. [36] subtracts a counterfactual baseline from Q function to highlight the individual's effect, facilitating rational independent reward allocation in multi-agent games. The Shapley value method is applied in [37], within the context of global reward games, to address the inaccurate allocation of contributions. On the basis of NE, many studies seek to find the Pareto optimal equilibrium for every agent involved [16], [19]. Ref. [16] connects the multiple gradient descent algorithm to the MARL, leading to strong Pareto optimal solutions. Recently, the PAC algorithm [19], which operates within a non-conflict game framework, assumes multi-agent interactions in an "optimistic" manner and ensures that all agents converge toward Pareto optimal equilibria.

We summarize the key differences between our algorithm and highly related literature in Table I.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In Section III-A, we describe the FL process and sequentially present the models of quantization, delay, and energy consumption. Subsequently, in Section III-B, we introduce the optimization problem, which aims to minimize network overhead while ensuring the convergence of FL training.

TABLE II
NOTATIONS MAINLY USED IN THIS PAPER

Notation	Description
$i \in [1, \cdots, N]$	Index of N FL clients
$r \in [1, \cdots, R]$	Number of SPs; also total number of agents
$T = [1, \cdots, t, \cdots]$	FL global training round
ι	Total steps of FL local update
$\mathcal{D}_{i,r}$	Client i's data set about service r
$\boldsymbol{\omega}, \boldsymbol{\omega}_{r,t}, \boldsymbol{\omega}_{i,r,t}^{(\iota)}$	FL model parameters
$L_r, L_{i,r}$	Global and local loss function of client i with respect to service r
$\mathrm{vol}_{i,r,t}, \mathrm{vol}_{r,t}$	Client i's communication overheads for service r in the t-th round; the combination of overhead $\mathrm{vol}_{i,r,t}$ from clients within service r
$T_{i,r,t}^{\mathrm{cmp}}, T_{i,r,t}^{\mathrm{com}}$	Client i's computational and communication delay at the t-th round with respect to service r
$E_{i,r,t}^{\mathrm{cmp}}, E_{i,r,t}^{\mathrm{com}}$	Client i's energy consumption at the t-th round with respect to service r
$v_{i,r,t}$	Client i's transmission rate of service r in the t-th round
$n_{r,t}$	SP r's client selection in the t-th round
$B_{i,r,t}, B_{r,t}$	Client i's bandwidth in the t-th round, the combination of selected clients' bandwidth
$\Gamma_{r,t}$	Model accuracy of service r in the t-th round
$q_{r,t}, q_{i,r,t}$	SP r's quantization decision in the t-th round, and the corresponding quantization level for client i
$f_{r,t}, f_{i,r,t}$	SP r's quantization decision in the t-th round, and the corresponding quantization level for client i
$o_{r,t}, a_{r,t}$	Agent r's observation and action in the t-th round
$Z_{r,t}, \Theta_{r,t}$	SP r's model training status in the t-th round
$\mathrm{rwd}_{r,t}$	Reward of agent r in the t-th round
J_r	Cumulative expected reward of agent r
M_r	Loss functions of agent r's value network
π_r,π_{-r},$\boldsymbol{\pi}^{\dagger}$	Agent r's policy, joint policy apart from agent r and conjectured joint policy
$V_r^{\boldsymbol{\pi}}$,$Q_r^{\boldsymbol{\pi}}$	State value function and Q-function of agent r

Beforehand, major notations in the paper are summarized in Table II.

A. System Model

1) FL Process: We primarily consider an FL scenario on shared network resources, as illustrated in Fig. 1, which

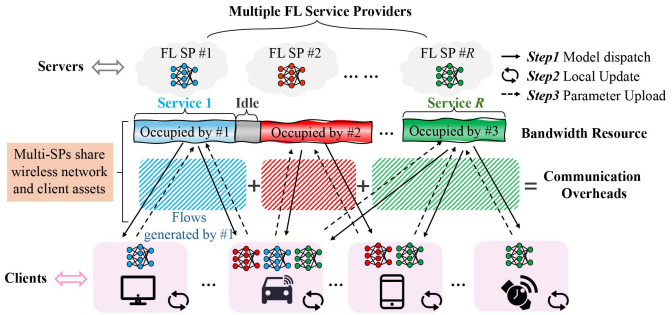

Fig. 1. A system overview of non-cooperative FL services with multiple SPs.

comprises a group of SPs $\mathcal{R} = \{1, \ldots, r, \ldots, R\}$ with an associated set of clients $\mathcal{N} = \{1, \ldots, i, \ldots, N\}$. Specifically, the server at each SP r is responsible for training a specific service model, while clients leverage their private data to provide local training, benefiting the learning of these models corresponding to different SPs. Without loss of generality, the local dataset at client i associated with service r is denoted as $\mathcal{D}_{i,r}$. Each sample $\xi \in \mathcal{D}_{i,r}$ consists of the feature set and its corresponding label. Thus, the local loss function for each client i can be defined as

$$L_{i,r}(\boldsymbol{\omega}) = \frac{1}{|\mathcal{D}_{i,r}|} \sum_{\xi \in \mathcal{D}_{i,r}} l(\boldsymbol{\omega}; \xi), \tag{1}$$

where the function $l(\boldsymbol{\omega}; \xi)$ measures the difference between predicted and true values based on the parameters $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_d, \ldots, \omega_{|\boldsymbol{\omega}|}]$ and the sampled data ξ. Correspondingly, the global minimization problem for service r can be defined as minimizing the aggregation of multiple local loss functions, namely,

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} L_r(\boldsymbol{\omega}) = \arg\min_{\boldsymbol{\omega}} \sum_{i=1}^{N} \kappa_{i,r} L_{i,r}(\boldsymbol{\omega}), \tag{2}$$

Typically, the aggregation weight $\kappa_{i,r} = |\mathcal{D}_{i,r}| / \sum_{i=1}^{N} |\mathcal{D}_{i,r}|$.

Considering the t-th global training round for service r, where the global model in (2) is represented as $\boldsymbol{\omega}_{r,t}$, the server distributes the model $\boldsymbol{\omega}_{r,t}$ to selected clients, who then perform ι steps' stochastic gradient descent (SGD)-based local updates to complete one full round of training. Mathematically, the local update method can be expressed as

$$\boldsymbol{\omega}_{i,r,t}^{(k)} = \boldsymbol{\omega}_{i,r,t}^{(k-1)} - \eta_k \tilde{\nabla} L_{i,r}\left(\boldsymbol{\omega}_{i,r,t}^{(k-1)}\right), \tag{3}$$

where $k \in \{1, \ldots, \iota\}$, η_k denotes a k-relevant learning rate, $\tilde{\nabla} L_{i,r}(\cdot)$ represents the stochastic gradient of the local loss function and $\boldsymbol{\omega}_{i,r,t}^{(0)} = \boldsymbol{\omega}_{r,t}$. After performing ι local updates, the client acquires a new model and uploads the quantization $\Psi(\boldsymbol{\omega}_{i,r,t}^{(\iota)})$ of corresponding parameters $\boldsymbol{\omega}_{i,r,t}^{(\iota)}$ to designated servers. On the server side, consistent with [38], these parameters uploaded by selected clients will be aggregated as

$$\boldsymbol{\omega}_{r,t+1} = \sum_{i=1}^{N} \kappa_{i,r} \Psi\left(\boldsymbol{\omega}_{i,r,t}^{(\iota)}\right). \tag{4}$$

The client uploads the model before compressing it according to the quantization policy $\Psi(\cdot)$ specified by the server. Notably, servers at different SPs can also calibrate suitable aggregation strategies, such as client selection, alongside effective resource allocation techniques, which will be discussed as follows.

2) Quantization Process: Following [39], we reduce the size of data flow by statistically quantizing elements of model parameters $\boldsymbol{\omega}$ into some discrete levels, with its d-th element's maximum-q-level quantization expressed as

$$\Psi_q(\omega_d) = \|\boldsymbol{\omega}\|_p \cdot \mathrm{sgn}(\omega_d) \cdot \Xi_q(\omega_d, q), \tag{5}$$

where $\|\cdot\|_p$ represents the p-norm, $\mathrm{sgn}(\cdot)$ denotes the sign function, and $\Xi_q(\omega_d, q)$ is a random mapping defined as

$$\Xi_q(\omega_d, q) = \begin{cases} u/q, & \varepsilon \le 1 - P\left(\frac{|\omega_d|}{\|\boldsymbol{\omega}\|_p}, q\right) \\ (u+1)/q, & \text{otherwise} \end{cases} \tag{6}$$

Here, ε denotes the probabilistic output of a uniformly distributed random variable. $P(e, q) = eq - u$, $0 \le e \le 1$, where u is any integer satisfying $u \in [0, q)$ and $\frac{u}{q} \le \frac{|\omega_d|}{\|\boldsymbol{\omega}\|_p} \le \frac{u+1}{q}$ [39]. Attributed to the value quantized by $\lceil \log_2 q \rceil + 1$ bits per element (including a one-bit sign), as well as a 32-bit gradient norm, the communication overheads for a $|\boldsymbol{\omega}|$-length quantized parameter vector that client i needs to upload for service r during the t-th round is reduced from the original $32|\boldsymbol{\omega}|$[1] to

$$\mathrm{vol}_{i,r,t} = |\boldsymbol{\omega}| \left(\lceil \log_2 q_{i,r,t} \rceil + 1\right) + 32, \tag{7}$$

where $q_{i,r,t}$ is the quantization level chosen by client i of the service r in the t-th round. Therefore, the total communication overheads for a single service is $\mathrm{vol}_{r,t} = \sum_{i \in \mathcal{N}} \mathrm{vol}_{i,r,t}$.

3) Communication Latency and Energy Consumption: In wireless network scenarios, FL training latency and total energy consumption primarily arise during computation and transmission of local model updates. The computational energy consumption for client i [20] during the t-th round training for service r can be expressed as

$$E_{i,r,t}^{\mathrm{cmp}} = \mu_i c_{i,r} |\mathcal{D}_{i,r}| f_{i,r,t}^2, \tag{8}$$

where μ_i is the effective capacitance constant determined by the chip architecture, $f_{i,r,t}$ denotes the CPU cycle frequency of client i in the t-th round, and $c_{i,r}$ represents the number of CPU cycles required to execute one sample for service r on client i. Concurrently, the incurred local computation latency can be written as

$$T_{i,r,t}^{\mathrm{cmp}} = c_{i,r} |\mathcal{D}_{i,r}| / f_{i,r,t}. \tag{9}$$

On the other hand, in the model parameter transmission phase, we consider frequency-division multiple access (FDMA) technology for communication between clients and SPs. Based on the Shannon formula, the transmission rate of client i can be denoted as

$$v_{i,r,t} = B_{i,r,t} \log_2 \left(1 + \frac{g_{i,t} p_{i,t}}{B_{i,r,t} N_0}\right), \tag{10}$$

where $B_{i,r,t}$ represent the service r-related bandwidth allocated to client i in the t-th global training round, $g_{i,t}$ and $p_{i,t}$ denote

[1]Each pre-quantization parameter ω_d is represented as 32-bit floating-point numbers, consistent with standard deep learning frameworks.

the channel gain and transmission power corresponding to client i, respectively, and N_0 is the single-sided white noise power spectral density. Therefore, the communication latency and transmission energy consumption of client i can be calculated as

$$T_{i,r,t}^{\mathrm{com}} = \frac{\mathrm{vol}_{i,r,t}}{v_{i,r,t}}, \tag{11}$$

$$E_{i,r,t}^{\mathrm{com}} = T_{i,r,t}^{\mathrm{com}} p_{i,t} = \frac{\mathrm{vol}_{i,r,t} p_{i,t}}{v_{i,r,t}}. \tag{12}$$

In summary, the total energy consumption and total latency of service r in the t-th round are expressed as follows

$$E_{r,t}^{\mathrm{total}} = \sum_{i \in \mathcal{N}} \left(E_{i,r,t}^{\mathrm{com}} + E_{i,r,t}^{\mathrm{cmp}} \right), \tag{13a}$$

$$T_{r,t}^{\mathrm{total}} = \max_{i \in \mathcal{N}} \left(T_{i,r,t}^{\mathrm{cmp}} + T_{i,r,t}^{\mathrm{com}} \right). \tag{13b}$$

B. Problem Formulation

Based on the communication and computation models above, each SP r independently aims to maximize its FL performance by optimizing client selection $n_{r,t}$, quantization $q_{r,t}$, and resource allocation ($B_{r,t} = \sum_{i \in \mathcal{N}} B_{i,r,t}$, $f_{r,t}$), balancing model accuracy, communication efficiency, and energy consumption over a finite horizon. This objective is formalized for each SP as

$$\begin{aligned} \min_{\boldsymbol{f},\boldsymbol{n},\boldsymbol{q},\boldsymbol{B}} \quad & \sum_{t=0}^{T-1} \gamma^t \Upsilon \left(L_r(\boldsymbol{\omega}_{r,t}), \mathrm{vol}_{r,t}, E_{r,t}^{\mathrm{total}}, T_{r,t}^{\mathrm{total}} \right) \\ \text{s.t.} \quad & \boldsymbol{C1} \quad E_{r,t}^{\mathrm{total}} \le E_r^{\max}, T_{r,t}^{\mathrm{total}} \le T_r^{\max}, \forall t, r \\ & \boldsymbol{C2} \quad n_{r,t} \in [1, N], \forall t, r \\ & \boldsymbol{C3} \quad f^{\min} \le f_{r,t} \le f^{\max}, \forall t, r \\ & \boldsymbol{C4} \quad B^{\min} \le \sum_{r} B_{r,t} \le B^{\max}, \forall t \\ & \boldsymbol{C5} \quad q_{r,t} \in [q^{\min}, q^{\max}], \forall t, r \end{aligned} \tag{14}$$

Here, γ denotes a discount factor while the monotonically decreasing function $\Upsilon(L_r(\boldsymbol{\omega}_{r,t}), \mathrm{vol}_{r,t}, E_{r,t}^{\mathrm{total}}, T_{r,t}^{\mathrm{total}})$ weights the components therein, whose exact format will be discussed subsequently. Constraints $\boldsymbol{C1}$, $\boldsymbol{C3}$, and $\boldsymbol{C4}$ impose limits on per-service energy consumption $E_r^{\max}$, total latency $T_r^{\max}$, computing capabilities $[f^{\min}, f^{\max}]$, and the overall bandwidth budget $[B^{\min}, B^{\max}]$ respectively, while $\boldsymbol{C2}$ and $\boldsymbol{C5}$ define the permissible ranges for client selection and quantization level. Implicitly, the shared constraint like $\boldsymbol{C4}$ underscores the inherent non-cooperative competition among SPs, where each seeks to maximize its individual resource acquisition for superior performance. Consequently, the systemic objective is to judiciously balance individual service quality with the achievement of a global equilibrium.

We reformulate (14) as a Markov decision process (MDP) for iterative optimization. To resolve the non-cooperative dynamics and achieve a robust system-wide equilibrium, we propose a game-theoretic Pareto actor-critic framework with expectile regression. This approach enables each SP to independently learn its optimal policy by conjecturing the joint policies of others, and accounting for heterogeneous risk preferences via expectile regression, ultimately guiding the system towards a global Pareto-optimal equilibrium.

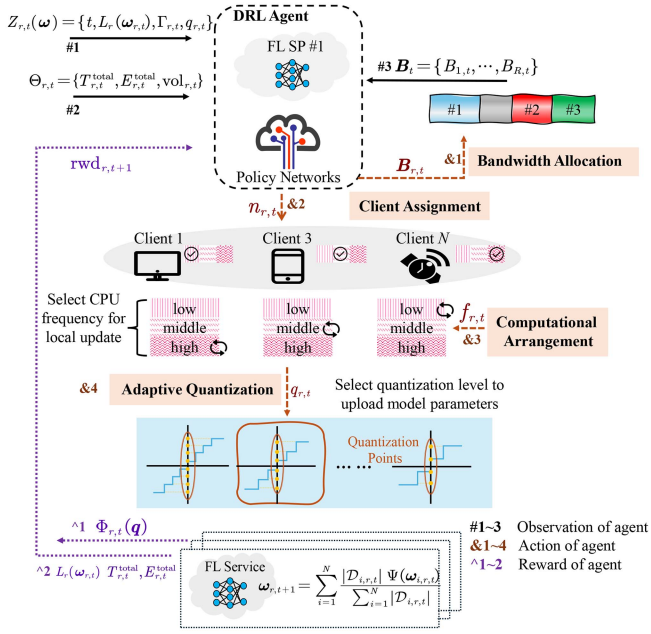

Fig. 2. The MDP diagram for a single FL service provider.

IV. PAC-MCoFL for Communication and Computation Co-Optimization

A. Markov Decision Process

We formulate the communication and computation co-optimization problem for FL with multi-SP as a sequential decision-making problem, modeled as an MDP. This MDP is represented by a quintuple $(\mathcal{O}, \mathcal{A}, \mathrm{rwd}, \Lambda, \gamma)$, where $\mathcal{O}$ and $\mathcal{A}$ denotes the observation and action space, while rwd is the reward function, obtained by mapping the current time state and action to a reward value. Λ indicates the state transition probability function, which captures the dynamics of the environment, and γ is a discount factor. The specific definitions of the observation, action, and reward are as follows.

- *Observation Space:* As shown in Fig. 2, we use the terminology *agent* to denote the intelligent decision-maker deployed at each SP. Consistent with the black solid line in Fig. 2, the observation of agent r after the t-th FL round can be represented as

$$o_{r,t} = \{ Z_{r,t}(\boldsymbol{\omega}), \Theta_{r,t}, \boldsymbol{B}_t \}, \tag{15}$$

where $Z_{r,t}(\boldsymbol{\omega}) = \{t, L_r(\boldsymbol{\omega}_{r,t}), \Gamma_{r,t}, q_{r,t}\}$ records the round time, loss, model accuracy and quantization level of FL training, $\Theta_{r,t} = \{T_{r,t}^{\mathrm{total}}, E_{r,t}^{\mathrm{total}}, \mathrm{vol}_{r,t}\}$ encompasses the current FL model training status, and $\boldsymbol{B}_t = \{B_{r,t}\}_{r \in \mathcal{R}}$ represents the bandwidth allocation for all SPs. This setting has practical significance. On one hand, since different SPs often resist publicly disclosing the specific operational

details, the first two elements, $\Theta_{r,t}$ and $Z_{r,t}$, contain internally available information only. On the other hand, to establish a game-theoretic relationship among agents in MARL, we deem the bandwidth allocation information as public information that can be acquired from the network operator [4].

- *Action Space:* As shown by the brown dashed line in Fig. 2, the action space, comprising all decision variables in (14), can be written as

$$a_{r,t} = \{n_{r,t}, f_{r,t}, B_{r,t}, q_{r,t}\} \in \mathcal{A}, \tag{16}$$

on which the underlying clients further determine their behavior. Specifically, the bandwidth $B_{r,t}$ is assumed to be equally allocated among the selected clients within each service, and the CPU frequency $f_{i,r,t}$ and quantization level $q_{i,r,t}$ for each client will fluctuate slightly under a given action [25], [40], i.e., $f_{i,r,t} \sim G(f_{r,t}, \Sigma_f), q_{i,r,t} \sim G(q_{r,t}, \Sigma_s)$, G is the Gaussian distribution. In addition, we denote $\boldsymbol{a}_t = [a_{1,t}, \ldots, a_{R,t}]$, and we assume the actions from other agents can be acquired from network operator and participated clients.

- *Reward Function:* To match the objective function Υ in (14), in line with the purple dotted line in Fig. 2, we define the reward function as

$$\mathrm{rwd}_{r,t} = \sigma_1 \Gamma_{r,t} + \sigma_2 \Phi_{r,t}(\boldsymbol{q}) - \sigma_3 E_{r,t}^{\mathrm{total}} - \sigma_4 T_{r,t}^{\mathrm{total}}, \tag{17}$$

where $\Gamma_{r,t}$ denotes test accuracy, the weights σ_j, $j \in \{1, \ldots, 4\}$, are positive constants, and the negative sign transforms the minimization problem of latency and energy consumption into a reward maximization problem. The reward function captures immediate performance metrics and actions. Through a single composite utility, the QoS of an SP, one type of highly sensitive information, has not been disclosed explicitly, reinforcing the non-cooperative nature of the problem. Besides, an adversarial factor Φ, which characterizes the quantization policy game across SPs to compete for communication resources, is defined as

$$\Phi_{r,t}(\boldsymbol{q}) = \frac{n_{r,t}\, q_{r,t}}{\epsilon \times \mathrm{vol}_{r,t} + \sum_{j \in \mathcal{R}/\{r\}} n_{j,t}\, q_{j,t}}, \tag{18}$$

where ϵ is a constant. Although FL convergence is inherently dependent on sequential state transitions, the discounted cumulative reward can propagate long-term dependencies via value function updates, enabling a tractable approximation for real-time co-optimization. Notably, when each agent makes the decision, it does not know the action concurrently taken by other agents. After each agent finally executes a determined action, the part $\sum_{j \in \mathcal{R}/\{r\}} n_{j,t}\, q_{j,t}$ can be known through the network operator's control plane, which is consistent with 3GPP-defined network exposure functions [41]. The communication overhead for action sharing is negligible compared to the transmission of FL model updates, as shown in Appendix D-A, available online.

With this formulated MDP, each SP executes an action based on its perceived environment, guided by a game-theoretic policy, and subsequently receives a corresponding reward. We will next discuss how to resort to the PAC algorithm to obtain a learned policy π_r parameterized by ϕ, which determines the action $a_{r,t}$ based on certain observations and strives to maintain service quality while minimizing communication and computation costs.

B. Expectile Regression-Based PAC for Multi-SP Game With Heterogeneous Risk Modeling

In this subsection, we present the algorithmic framework of PAC-MCoFL. We first define the joint policy $\boldsymbol{\pi} = (\pi_r, \pi_{-r})$ where π_{-r} represents the set of policies from all other agents (denoted as $-r$). In addition, there is a finite time horizon structure for RL in PAC-MCoFL. For each SP r, we denotes the cumulative expected reward under joint policy $\boldsymbol{\pi}$ as

$$J_r(\boldsymbol{\pi}) = \mathbb{E}_{o_{r,0}}\left[V_r^{\boldsymbol{\pi}}(o_{r,0})\right] = \mathbb{E}_{o_{r,0}}\left[\sum_{t=0}^{T-1} \gamma^t \mathbb{E}_{\boldsymbol{\pi}}\left[\mathrm{rwd}_{r,t} | o_{r,0}, \boldsymbol{\pi}\right]\right], \tag{19}$$

where $V_r^{\boldsymbol{\pi}}$ is state value function of agent r under joint policy $\boldsymbol{\pi}$, and $\gamma \in [0, 1)$ denotes the discount factor. Or equivalently,

$$J_r(\boldsymbol{\pi}) = \sum_{o_{r,t}} d^{\boldsymbol{\pi}}(o_{r,t})\left[V_r^{\boldsymbol{\pi}}(o_{r,t})\right], \tag{20}$$

where $d^{\boldsymbol{\pi}}(o)$ is the transfer probability from initial observation $o_{r,0}$ to observation o in the smooth distribution of the MDP under $\boldsymbol{\pi}$. Besides, the Q-function can be expressed as $Q_r^{\boldsymbol{\pi}}(o_{r,t}, \boldsymbol{a}_t) = \mathrm{rwd}_{r,t}(o_{r,t}, \boldsymbol{a}_t) + \gamma \mathbb{E}[V_r^{\boldsymbol{\pi}}(o_{r,t+1})]$.

A Pareto-optimal equilibrium [19] is a state where no agent can enhance its expected return without reducing the expected return of another agent. Drawing inspiration from the prisoner's dilemma [42], if agents can coordinate effectively and trust that neither will deviate from the equilibrium, the resulting joint policy not only maximizes the expected return for an individual agent but also for all agents involved. In other words, for agent r, if the other agents adhere $\pi_{-r}^{\dagger}$ maximizing $J_r(\pi_r, \pi_{-r})$ as

$$\pi_{-r}^{\dagger} = \arg\max_{\pi_{-r}} J_r\left(\pi_r, \pi_{-r}\right), \tag{21}$$

it is feasible for the PAC-based agent to learn a policy π_r that maximises $J_r(\pi_r, \pi_{-r}^{\dagger})$. This process culminates in a joint policy $\boldsymbol{\pi}^{\dagger} = (\pi_r, \pi_{-r}^{\dagger})$ which maximizes the cumulative rewards of all agents and eventually replaces NE with Pareto optimal equilibrium [19] as

$$\pi_r = \arg\max_{\pi_r} J_r\left(\pi_r, \pi_{-r}^{\dagger}\right). \tag{22}$$

Therefore, the difficulty turns to be conjecturing a joint policy $\pi_{-r}^{\dagger}$, which in practice, can be approximated by evaluating all potential joint actions of the other agents (i.e. all $a_{-r,t} \in \mathcal{A}_{-r}, \mathcal{A}_{-r} = \bigcup_{j \neq r} \mathcal{A}_j$) on the joint Q-value and taking the maximum operator, formally,

$$\pi_{-r}^{\dagger} = \arg\max_{a_{-r,t}} Q_r^{\boldsymbol{\pi}^{\dagger}}\left(o_{r,t}, a_{r,t}, a_{-r,t}\right). \tag{23}$$

In classic RL, the loss function for Q-function updates is defined by the temporal difference (TD) error in (24) shown at the bottom of this page. Nevertheless, considering heterogeneous risk preferences among SPs, such as spanning conservative to aggressive policies, we employ expectile regression [17] for critic loss computation

$$M_r^{(\tau)}(\boldsymbol{\pi}) \triangleq \mathbb{E}_{a_{r,t},a_{r,t+1}\sim\pi_r,a_{-r,t}\sim\pi_{-r}}\left[\tau\cdot\delta_+^2+(1-\tau)\cdot\delta_-^2\right], \tag{26}$$

where $\tau\in(0,1)$ is adjustable expectile factor controlling sensitivity to overestimation ($\tau<0.5$, aggressive) or underestimation ($\tau>0.5$, conservative); δ_+ and δ_- denote the positive and negative components of δ, respectively. By tuning τ, the critic flexibly characterizes the differential risk preferences of SPs toward policy decisions. Accordingly the critic is updated by

$$\theta_r=\theta_r-\alpha\nabla_{\theta_r}M_r^{(\tau)}(\boldsymbol{\pi}), \tag{27}$$

where α denotes the learning rate for updates. On the other hand, to maximize the expected discounted cumulative reward, the policy network can be updated by

$$\phi_r=\phi_r+\zeta\nabla_{\phi_r}J_r(\boldsymbol{\pi}), \tag{28}$$

where the policy gradient can be computed by taking the derivative of ϕ as shown in (25) at the bottom of this page with a learning rate ζ. As a side note, a baseline is commonly subtracted from $J_r(\boldsymbol{\pi})$ without altering the gradient computation but reducing the bias of the gradient estimate [13], [43]. We can compute this baseline through $\sum_{a_{r,t}}\pi_r(a_{r,t}|o_{r,t};\phi_r)Q_r^{\boldsymbol{\pi}^\dagger}(o_{r,t},a_{r,t},a_{-r,t})$ to avoid training an additional state value network.

Overall, compared to the existing architecture [13], [43], PAC-MCoFL benefits from PAC, which conjectures other agents' policies, to facilitate the determination of co-optimization strategies, including quantization levels, resource allocation, and client assignment. Therefore, PAC-MCoFL has the advantage of preventing the game from easily falling into suboptimal solutions. In summary, the pseudocode for PAC-MCoFL is summarized in the Algorithm 1.

C. TCAD for Dimension Reduction

The inherent complexity of multi-dimensional action spaces in non-cooperative multi-SP FL poses significant challenges for conventional MARL frameworks, due to the curse of dimensionality [44]. Additionally, independent action selection often result in coordination blindness, failing to capture critical interdependencies [9]. Therefore, we propose TCAD, a hierarchical action decomposition mechanism enabling efficient exploration in high-dimensional spaces while preserving cross-variable dependencies.

Let the hybrid action vector of SP r be defined as $a_{r,t}=\{n_{r,t},f_{r,t},B_{r,t},q_{r,t}\}$. We establish a ternary projection operator $\mathcal{T}_{\text{TCAD}}$ and the action space decomposition is formalized through

$$\mathcal{T}_{\text{TCAD}}: a_{r,t}\to\{\varsigma_m\cdot\psi_m\}_{m=1}^4,\quad \psi_m=\{-1,0,1\}, \tag{29}$$

where m represents the dimensionality, ς_m denotes the step-size granularity, which is a pre-defined hyperparameter tailored to each variable's physical constraints (see Appendix C-C, available online for related hyperparameter evaluation). The discrete-continuous hybrid update rules are unified as

$$a_{r,t+1}^{\prime(m)}=\text{Proj}_{\mathcal{C}_m}(a_{r,t}^{\prime(m)}+\psi_m\cdot\varsigma_m), \tag{30}$$

where $\text{Proj}_{\mathcal{C}_m}(\cdot)$ denotes the projection operator enforcing domain constraints

$$\mathcal{C}_m=\begin{cases}\{1,\dots,N\} & m=n\\ \left[f^{\min},f^{\max}\right] & m=f\\ \left[B^{\min},B^{\max}\right] & m=B\\ \mathbb{Z}\cap\left[q^{\min},q^{\max}\right] & m=q\end{cases} \tag{31}$$

The Cartesian reconstruction $\mathcal{A}'=\prod_{m=1}^4\{-1,0,1\}$ reduces the joint action space cardinality from $O(N\cdot\frac{f^{\max}-f^{\min}}{\varsigma_f}\cdot\frac{B^{\max}-B^{\min}}{\varsigma_B}\cdot q^{\max})$ to constant complexity, facilitating tractable computation of (23) and enabling fine-grained multi-dimensional control. The computational efficacy analysis of TCAD can be found in Appendix C-B, available online.

D. Parameterized Conjecture Generator

The original conjecture mechanism in (23) requires exhaustive search over opponents' joint action space $\mathcal{A}_{-r}$, incurring computational complexity of $O(|\mathcal{A}|^{R-1})$ for R SPs. To overcome this combinatorial bottleneck, we introduce PAC-MCoFL-p, a variant of PAC-MCoFL with an extra, φ_r-parameterized generator $\text{Gen}_{\varphi_r}(\cdot)$ for policy conjecture.

$$M_r(\boldsymbol{\pi})\triangleq\mathbb{E}_{a_{r,t},a_{r,t+1}\sim\pi_r,a_{-r,t}\sim\pi_{-r}}\left[\left(\underbrace{\text{rwd}_{r,t}+\gamma\max_{a_{-r,t}}Q_r^{\boldsymbol{\pi}^\dagger}(o_{r,t+1},a_{r,t+1},a_{-r,t})-Q_r^{\boldsymbol{\pi}^\dagger}(o_{r,t},a_{r,t},a_{-r,t})}_{\text{TD error term }\delta}\right)^2\right] \tag{24}$$

$$\begin{aligned}\nabla_{\phi_r}J_r(\boldsymbol{\pi}) &= \sum_{o_{r,t}}d^{\boldsymbol{\pi}}(o_{r,t})\sum_{a_{r,t},a_{-r,t}}\nabla_{\phi_r}\pi^\dagger(\boldsymbol{a}_t\mid o_{r,t})Q_r^{\boldsymbol{\pi}^\dagger}(o_{r,t},a_{r,t},a_{-r,t})\\ &= \sum_{o_{r,t}}d^{\boldsymbol{\pi}}(o_{r,t})\sum_{a_{r,t},a_{-r,t}}\pi_{-r}^\dagger(a_{-r,t}\mid o_{r,t},a_{r,t})\nabla_{\phi_r}\pi_r(a_{r,t}\mid o_{r,t})Q_r^{\boldsymbol{\pi}^\dagger}(o_{r,t},a_{r,t},a_{-r,t})\\ &= \sum_{o_{r,t}}d^{\boldsymbol{\pi}}(o_{r,t})\sum_{a_{r,t},a_{-r,t}}\pi_r(a_{r,t}\mid o_{r,t})\pi_{-r}^\dagger(a_{-r,t}\mid o_{r,t},a_{r,t})\cdot\nabla_{\phi_r}\log\pi_r(a_{r,t}\mid o_{r,t})Q_r^{\boldsymbol{\pi}^\dagger}(o_{r,t},a_{r,t},a_{-r,t})\end{aligned} \tag{25}$$

Algorithm 1: Training Process of the PAC-MCoFL Algorithm From the Perspective of Agent (i.e., SP) r.

Input: Total episodes $\text{num}_{\max}$; initialized parameters of the actor ϕ_r and critic θ_r of the agent r; FL global model $\boldsymbol{\omega}_r$, FL global training rounds T; total client set $\mathcal{N}$.

Output: Trained parameters of the actor ϕ_r and critic θ_r of the agent r.

Load the training dataset of FL service; $\text{num}_{\text{eps}} \leftarrow 0$;
while *each episode* $\text{num}_{\text{eps}} \leq \text{num}_{\max}$ **do**
  Initialize global model parameters $\boldsymbol{\omega}_{r,0}$ and client features, e.g., $|\mathcal{D}_{i,r}|, o_{r,0}, f_{r,0}$;
  for *each round* $t = 0, 1, \cdots, T-1$ *in* **do**
    Get observation $o_{r,t}$ and $o_{-r,t}$;
    Sample action delta $\{\psi_m\}_{m=1}^{4}$ from actor network π_r, where $\psi_m \in \{-1, 0, 1\}$;
    Construct full action $a_{r,t} = \{n_{r,t}, f_{r,t}, B_{r,t}, q_{r,t}\}$ by TCAD update rule from Eq. (30);
    for *each selected client* $i \in \mathcal{N}$ **do**
      Local update by Eq. (3) and get $\boldsymbol{\omega}_{i,r,t}$;
      Obtain $E^{\text{cmp}}_{i,t,r}$, $T^{\text{cmp}}_{i,t,r}$ by Eq. (8) and (9);
      Obtain $E^{\text{com}}_{i,t,r}$, $T^{\text{com}}_{i,t,r}$ by Eq. (12) and (11);
    end
    Global update $\boldsymbol{\omega}_{r,t+1} = \sum_{i=1}^{N} \frac{|\mathcal{D}_{i,r,t}| \Psi(\boldsymbol{\omega}_{i,r,t})}{\sum_{i=1}^{N} |\mathcal{D}_{i,r,t}|}$;
    Distributes global model $\boldsymbol{\omega}_{r,t+1}$ to all clients;
    Obtain $\Gamma_{r,t}$ through model test;
    Obtain $E^{\text{total}}_{t,r}$, $T^{\text{total}}_{t,r}$ by Eq. (13);
    Get $o_{r,t+1}, o_{-r,t+1}$ by Eq. (15);
    Compute $\text{rwd}_{r,t}$ by Eq. (17);
    Put $(o_{r,t}, a_{r,t}, \text{rwd}_{r,t}, o_{r,t+1})$ into an episode batch.
  end
  Insert the episode batch into a replay buffer H;
  Clear the episode batch.
  if H *reaches the buffer size for training* **then**
    Sample a single batch from H;
    Obtain Conjectured joint policy for other agents $\pi^{\dagger}_{-r} = \arg\max_{a_{-r,t}} Q_r^{\boldsymbol{\pi}^{\dagger}}(o_{r,t}, a_{r,t}, a_{-r,t}), \forall t$;
    //Note: For the scalable PAC-MCOFL-p*, this brute-force search is replaced by the parameterized generator as Eq.* (32);
    Update critic parameters θ_r by performing a gradient step to minimize the expectile loss $M_r^{(\tau)}$ from Eq (27);
    Update actor parameters ϕ_r by ascending the policy gradient $\nabla_{\phi_r} J_r$ from Eq. (28), which is conditioned on the conjectured policy $\pi^{\dagger}_{-r}$;
  end
  $\text{num}_{\text{eps}} \longleftarrow \text{num}_{\text{eps}} + 1$.
end

Specifically, PAC-MCoFL-p directly produces optimal joint actions of other agents through

$$\tilde{\pi}^{\dagger}_{-r} \triangleq \text{softmax}\left(\text{Gen}_{\varphi_r}(a_{r,t}, o_{r,t}, h_{-r,t})\right), \tag{32}$$

where $h_{-r,t}$ is the hidden state derived from $o_{-r,t}$. The generator is optimized through a compound loss function in (36), where χ is constant coefficient, D_{KL} denotes KL-divergency, $\pi^{\dagger,\text{tar}}_{-r}$ represents moving average of other agents' empirical policy distributions. This formulation ensures the conjectured actions maintain temporal consistency with historically optimal behavior while maximizing expected return. We enable end-to-end training of the generator by incorporating the aforementioned loss into the policy gradient in (28). The theoretical justification for this approach is provided by Theorem 1.

Theorem 1: Consider the scenario where the true optimal joint action distribution, denoted as $\pi^{\dagger}_{-r}$ (from (23)), is approximated by a parameterized Gen_{φ_r} yielding distribution $\tilde{\pi}^{\dagger}_{-r}$. Under Lipschitz continuity of Q-function given in Assumption 4, the expected Q-value error for the parameterized policy satisfies

$$|\mathbb{E}_{\tilde{\pi}^{\dagger}_{-r}}\left[Q_r^{\boldsymbol{\pi}^{\dagger}}\right] - \max_{a_{-r,t}} Q_r^{\boldsymbol{\pi}^{\dagger}}| \leq C \cdot \sqrt{D_{\text{KL}}\left(\tilde{\pi}^{\dagger}_{-r} || \pi^{\dagger}_{-r}\right)}, \tag{33}$$

where C denotes the Lipschitz constant of the Q-function.

We leave Assumption 4 and the proof in Appendix A-A, available online. This theorem shows that PAC-MCOFL-p is not merely a heuristic but a principled approximation with a provably bounded error. This justifies its application to large-agent scenarios where an exhaustive search is intractable, a finding further corroborated by our experiments.

E. Proof of Convergence

We first consider the case without expectile regression, where the critic loss is given by (24). Afterward, we will extend the result to the variant with expectile regression. In the former case, the update rule for the weight adjustment in (27) is derived from an iterative update [15], denoted as (34), on which the action value function $Q_{r,t}$ is iterated.[2] The update of the Q-function through PAC, specifically the inclusion of additional, conjectured action inputs from other agents based on the max operation, results in a significantly expanded exploration space and compromises existing conclusions. Consequently, the convergence dynamics that are missing in the classical PAC framework [19] are worthy of further investigation. To prove that the joint Q-function $\boldsymbol{Q}_t = [Q_{1,t}, \ldots, Q_{R,t}]$ converges gradually to Nash Q-value $\boldsymbol{Q}^* = [Q_1^*, \ldots, Q_R^*]$, we first define a Pareto operator as (35) on a complete metric space, as shown in Lemma 1.

Lemma 1: Considering any two action value functions $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{O}||\mathcal{A}|}$ of a single agent, for Pareto operator in (35), the following contraction mapping holds

$$\|\mathscr{H}^P Q_1 - \mathscr{H}^P Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty, \tag{38}$$

where $\|\cdot\|_\infty$ represents the supremum norm, $\gamma \in [0, 1)$. Besides, Pareto operator $\mathscr{H}^P$, which constitutes a contraction mapping, has the fixed point at $\boldsymbol{Q}^*$.

We leave the proof in Appendix A-B, available online.

Hu et al. [34] define an iterative process for obtaining Nash strategies through the Lemke-Howson algorithm and introduce a Nash operator $\mathscr{H}^N$.

[2] In the rest of the paper, the superscript $\boldsymbol{\pi}$ for Q and V will be omitted for simplicity of representation.

Definition 1 (Nash Operator): The expression for value function iteration is given by a Nash operator as

$$\mathscr{H}^N \boldsymbol{Q}(o,\boldsymbol{a}) = \mathbb{E}_{o_{t+1}\sim\Lambda(\cdot|o,\boldsymbol{a})} \left[\mathrm{rwd}(o,\boldsymbol{a}) + \gamma \boldsymbol{V}^{\mathrm{Nash}}(o_{t+1})\right]. \tag{39}$$

Definition 2 (Nash Equilibrium): In a stochastic game, a joint policy $\boldsymbol{\pi}^* \triangleq [\pi_1^*, \ldots, \pi_R^*]$ constitutes an NE if no agent can achieve a higher return by unilaterally changing its policy. Formally, it satisfies

$$\forall o_r \in \mathcal{O}_r,\ V_r(o_r;\boldsymbol{\pi}^*) = V_r\left(o_r;\pi_r^*,\pi_{-r}^*\right) \geq V_r\left(o_r;\pi_r,\pi_{-r}^*\right). \tag{40}$$

Therefore, by the proof of Lemma 1, the Nash operator constructs a contraction mapping on a complete metric space, ensuring that the Q-function ultimately converges to the value corresponding to the NE of the game, meaning there exists a fixed point $\mathscr{H}^N \boldsymbol{Q}^* = \boldsymbol{Q}^*$[34].

Then, we find the update rule for the Q-function in (34) follows a stochastic approximation structure. By Theorem 1 in [45] and Corollary 5 in [46], the stochastic approximation has the following convergence properties.

Lemma 2: The random process $\{\Delta_t\}$ defined in $\mathbb{R}$ as

$$\Delta_{t+1}(x) = (1-\alpha_t(x))\,\Delta_t(x) + \alpha_t(x)F_t(x),$$

converges to zero with probability 1 (w.p.1) when

1) $0 \leq \alpha_t(x) \leq 1, \sum_t \alpha_t(x) = \infty, \sum_t \alpha_t^2(x) < \infty$;
2) $x \in \mathscr{X}$, the set of possible states, and $|\mathscr{X}| < \infty$;
3) $\|\mathbb{E}[F_t(x) \mid \mathscr{F}_t]\|_W \leq \gamma\|\Delta_t\|_W + c_t$, where $\gamma \in [0,1)$ and c_t converges to zero w.p.1;
4) $\mathrm{var}[F_t(x) \mid \mathscr{F}_t] \leq U(1 + \|\Delta_t\|_W^2)$ with constant $U > 0$.

Here $\mathscr{F}_t$ denotes the filtration of an increasing sequence of σ-fields including the history of processes; $\alpha_t, \Delta_t, F_t \in \mathscr{F}_t$ and $\|\cdot\|_W$ is a weighted maximum norm.

To mimic the structure of Lemma 2, we can subtract Q_r^* from both sides of (34) and define

$$\Delta_t(x) = Q_{r,t}(o_{r,t},\boldsymbol{a}_t) - Q_r^*(o_{r,t},\boldsymbol{a}_t), \tag{41}$$

$$F_t(x) = \mathrm{rwd}_{r,t} + \gamma \max_{a_{-r,t}} Q_{r,t}(o_{r,t+1}, a_{r,t+1}, a_{-r,t}) - Q_r^*(o_{r,t},\boldsymbol{a}_t), \tag{42}$$

where $(o,\boldsymbol{a}) = x$ represents the visited observation-action pairs, and $\alpha_t(x)$ is interpreted as the learning rate. The Q-function is updated by each agent using only the pairs $(o_{r,t},\boldsymbol{a}_t)$ at the time t when they are accessed. In other words, $\alpha_t(o',\boldsymbol{a}') = 0$ for any $(o',\boldsymbol{a}') \neq (o_{r,t},\boldsymbol{a}_t)$.

Besides, we introduce several key assumptions.

Assumption 1: The observation space $\mathcal{O}$ and action space $\mathcal{A}$ of the MDP are finite; any observation-action pair $(o,\boldsymbol{a})$ can be accessed an infinite number of times.

Assumption 2: The reward function is independent of the next state, and the reward function is bounded.

Assumption 3: For the joint action value function $\boldsymbol{Q}_t$ at any stage of the game, the NE policy $\boldsymbol{\pi}^*$ can only be 1) a global optimum or 2) a saddle point. Moreover, both cases share the same Nash value (Ω is the policy space), namely,

1)
$$\mathbb{E}_{\boldsymbol{\pi}^*}\left[Q_{r,t}(o_{r,t},\boldsymbol{a}_t)\right] \geq \mathbb{E}_{\boldsymbol{\pi}}\left[Q_{r,t}(o_{r,t},\boldsymbol{a}_t)\right], \forall \boldsymbol{\pi} \in \bigcup_r \Omega_r; \tag{43}$$

2)
$$\mathbb{E}_{\boldsymbol{\pi}^*}\left[Q_{r,t}(o_{r,t},\boldsymbol{a}_t)\right] \geq \mathbb{E}_{\pi_r}\mathbb{E}_{\pi_{-r}^*}\left[Q_{r,t}(o_{r,t},\boldsymbol{a}_t)\right], \forall \pi_r \in \Omega_r \tag{44}$$

and

$$\mathbb{E}_{\boldsymbol{\pi}}\left[Q_{r,t}(o_{r,t},\boldsymbol{a}_t)\right] \leq \mathbb{E}_{\pi_r^*}\mathbb{E}_{\pi_{-r}}\left[Q_{r,t}(o_{r,t},\boldsymbol{a}_t)\right], \quad \forall \pi_{-r} \in \bigcup_{j\neq r} \Omega_j. \tag{45}$$

Assumption 1 (finite actions) is inherently satisfied by our TCAD mechanism (Section IV-C), which explicitly discretizes the multi-dimensional action space. Assumption 2 (bounded rewards) holds because our reward function, i.e. (17), is a composition of physically bounded metrics, while Assumption 3 is a standard requirement in multiagent equilibrium analyses [14], [15], [34]. Finally, the convergence theorem without involving expectile regression can be summarized as follows.

Theorem 2: In a multi-agent stochastic game setting, under Assumptions 1, 2, and 3, as well as the first condition of Lemma 2, the values updated iteratively through (34) will ultimately converge to an NE point, specifically the Nash Q-value $\boldsymbol{Q}^* = [Q_1^*, Q_2^*, \ldots, Q_R^*]$.

We leave the proof in Appendix A-C, available online. Next, considering a Pareto operator with expectile regression defined as (37) shown at the bottom of this page, the same convergence result can be transferred accordingly.

Corollary 1: Let the assumptions of Theorem 2 hold. Suppose each agent $r \in \mathcal{R}$ employs an expectile coefficient $\tau_r \in (0,1)$, the Q-function sequence generated by a modified Pareto operator maintains the almost sure convergence properties established in Theorem 2.

We give the proof in Appendix A-D, available online. This theoretical guarantee of convergence is empirically validated in

$$Q_{r,t+1}(o_{r,t},\boldsymbol{a}_t) = Q_{r,t}(o_{r,t},\boldsymbol{a}_t) + \alpha\left(\mathrm{rwd}_{r,t} + \gamma\mathbb{E}_{o_{r,t+1},a_{r,t+1}}\left[\max_{a_{-r,t}} Q_{r,t}(o_{r,t+1},a_{r,t+1},a_{-r,t})\right] - Q_{r,t}(o_{r,t},\boldsymbol{a}_t)\right) \tag{34}$$

$$\mathscr{H}^p Q_{r,t}(o_{r,t},\boldsymbol{a}_t) = \mathbb{E}_{o_{r,t+1}\sim\Lambda(\cdot|o_{r,t},\boldsymbol{a}_t)}\left[\mathrm{rwd}_{r,t} + \gamma \max_{a_{-r,t}} Q_{r,t}(o_{r,t+1},a_{r,t},a_{-r,t})\right] \tag{35}$$

$$\Phi(\varphi_r) = \mathbb{E}_{\tilde{a}_{-r,t}\sim\tilde{\pi}_{-r}^\dagger}\left[-Q_r^{\boldsymbol{\pi}^\dagger}(o_{r,t},a_{r,t},\tilde{a}_{-r,t})\right] + \chi\mathbb{E}\left[D_{\mathrm{KL}}\left(\tilde{\pi}_{-r}^\dagger || \pi_{-r}^{\dagger,\mathrm{tar}}\right)\right] \tag{36}$$

$$\mathscr{H}^{p,\tau} Q_{r,t}(o_{r,t},\boldsymbol{a}_t) = \mathbb{E}_{o_{r,t+1}\sim\Lambda}\left[\mathrm{rwd}_{r,t} + \gamma\tau\max_{a_{-r,t}} Q_{r,t}^+(o_{r,t+1},a_{r,t},a_{-r,t}) + \gamma(1-\tau)\max_{a_{-r,t}} Q_{r,t}^-(o_{r,t+1},a_{r,t},a_{-r,t})\right] \tag{37}$$

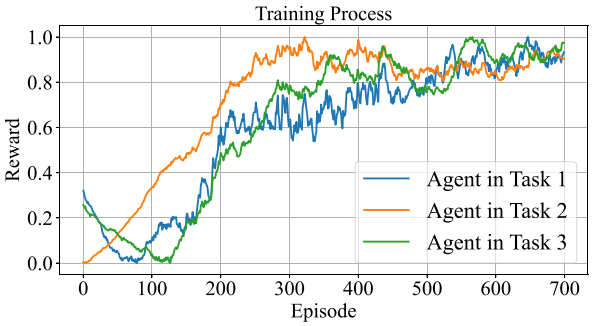

Fig. 3. Agents' training process, where rewards are normalized for display purposes.

our experiments (e.g., Fig. 3), where the gradually stabilized reward curves confirm the practical reliability.

V. Simulation Settings and Results

A. Default Simulation Settings

We consider an FL environment involving five clients (i.e., $N = 5$), coordinated by three SPs (i.e., $R = 3$). These SPs schedule the clients to train three distinct tasks with CIFAR-10 [Task r_1, high complexity/50,000 RGB images/10 classes], FashionMNIST [Task r_2, medium complexity/70,000 grayscale images/10 classes], and MNIST [Task r_3, low complexity/60,000 grayscale images/10 classes] datasets, respectively. This benchmark selection, across varying complexities, evaluates our algorithm's efficacy in a multi-SP FL setting, consistent with common/mainstream edge intelligence applications [20], [47]. Experiments are performed on an Ubuntu 20.04 workstation equipped with an NVIDIA GeForce RTX 3090 GPU and AMD Ryzen 9 7950X 16-Core CPU, using Python, CUDA 12.2, and PyTorch 2.5. Accordingly, three image classification models are leveraged. For Task r_1, we apply a convolutional neural network (CNN) with four convolutional layers, followed by three fully connected layers. The model for Task r_2 is a shallow CNN, with two convolutional layers followed by two fully connected layers. The model for Task r_3 only contains two fully connected layers. The number of model parameters for three tasks are 9 074 474, 21 840 and 101 770, respectively. The datasets are distributed among the clients in an independent and identically distributed (IID) manner (i.e., a non-IID degree of $\rho = 1$), indicating each client receives data covering 100% of the label categories. FL training utilizes the Adam optimizer with learning rate $\eta_k = 0.001$, $k \in \{1, \ldots, \iota\}$, local iterations $\iota = 3$, and a mini-batch size of 64. Each simulation episode spans up to $T = 35$ FL rounds.[3] We use the reward function in (17), along with its associated metrics, to evaluate the performance of a particular SP. Within our algorithm, the critic network employs two fully connected layers of 64 and 128 neurons to estimate state–action values, while the actor network uses three fully connected layers of 64, 128, and 64 neurons to map observations to actions. We consider the following comparison baselines:

- *FedAvg:* Vanilla FL [38] for each SP without quantization/resource optimization.
- *FedProx-u:* Uniform 8-bit quantization variants of [48] with equal resource allocation.
- *FedDQ-h & AdaQuantFL-h:* Adaptive quantization policy during FL training [24], [25], with heuristic computation and communication resource allocation rules tied to quantization levels.
- *MAPPO:*[4] Resource scheduling and FL operation management based on the MAPPO [13], where no conjectured joint policy is introduced. MAPPO is specifically included as the most relevant RL competitor given its proven effectiveness in FL resource allocation [7], [44].
- *RSM-MASAC:* Resource scheduling and FL operation management based on the RSM-MASAC [49], an enhanced variant of multi-agent soft actor–critic (MASAC).

These baseline algorithms collectively cover essential optimization dimensions in multi-SP FL scenarios—including conventional and adaptive quantization, uniform and heuristic resource allocation, and MARL-based global scheduling—thereby enabling a comprehensive evaluation of the performance impact. Finally, main simulation parameters are summarized in Table III.

To comprehensively compare the performance in the non-cooperative game setting, besides metrics like individual and

TABLE III
Simulation Parameters

Parameters	Descriptions	Values
N	Number of clients	5
R	Number of tasks	3
ρ	Degree of non-IID data	1
ι	FL local update steps	3
τ	Expectile factor	0.5
T	FL global training rounds	35
$g_{i,t}$	Channel gain	$[-63, -73]$ dB
N_0	Noise power spectral density	$[-124, -174]$ dBm/Hz
$p_{i,t}$	Clients' transmission power	$[10, 33]$ dBm
μ_i	Effective switching capacitance constant	10^{-27}
$c_{i,1}, c_{i,2}$	CPU cycles consumed per sample for r_1, r_2	$[6.07, 7.41] \times 10^5$
$c_{i,3}$	CPU cycles consumed per sample for r_3	$[1.10, 1.34] \times 10^8$
σ_1	Weighting factor 1 for r_1, r_2, r_3	100, 100, 100
σ_2	Weighting factor 2 for r_1, r_2, r_3	4.8, 31.25, 12.5
σ_3, σ_4	Weighting factor 3 and 4 for r_1, r_2, r_3	0.8, 25, 16.6
ζ, α	Learning rate	0.001, 0.001
$\varsigma_n, \varsigma_q, \varsigma_f, \varsigma_B,$	TCAD Granularity	1,4,0.5,2
Σ_q, Σ_f	Jitter factor for quantization and CPU frequency	0.25, 0.5
$f^{\min}, f^{\max}$	CPU frequency range	$[0.5, 3.5]$ GHz
$q^{\min}, q^{\max}$	Quantization level range	$[2, 32]$ Bits
$B^{\min}, B^{\max}$	SP's bandwidth range	$[0, 30]$ MHz

[3] In each round t, the agent observes the state $o_{r,t}$, conjectures the optimal opponent action $\pi^{\dagger}_{-r}$ that maximizes its Q-value, and then selects its own action $a_{r,t}$ via its policy network. Upon execution of the joint action, the environment returns a reward $\mathrm{rwd}_{r,t}$ and the next state $o_{r,t+1}$. This complete transition is then stored in a replay buffer for RL training.

[4] The implementation details of MARL solutions are given in Appendix B-C, available online.

TABLE IV
PERFORMANCE EVALUATION OF MULTIPLE FL SPS FRAMEWORKS

Metric	Fixed	Uniform	Heuristic		MARL			
	FedAvg	FedProx-u	FedDQ-h	AdaQuantFL-h	MAPPO	RSM-MASAC	PAC-MCoFL	PAC-MCoFL-p
$\overline{\text{vol}}_{1,t}$	65.33($\pm$3.86)	27.22($\pm$4.32)	36.3($\pm$5.87)	**8.97**($\pm$2.73)	23.01($\pm$4.86)	20.65($\pm$3.86)	18.74($\pm$3.23)	22.02($\pm$2.08)
$\overline{T}_{1,t}$	46.81($\pm$6.06)	34.44($\pm$7.22)	15.82($\pm$4.21)	9.23($\pm$2.36)	11.62($\pm$3.06)	4.94($\pm$2.34)	6.62($\pm$1.26)	**4.26**($\pm$2.34)
$\overline{E}_{1,t}$	12.62($\pm$3.41)	8.01($\pm$1.21)	19.17($\pm$3.22)	1.84($\pm$0.14)	11.28($\pm$2.02)	**4.12**($\pm$1.12)	9.96($\pm$2.12)	9.61($\pm$1.87)
$\overline{\text{rwd}}_{1,t}$	48.92($\pm$9.67)	65.41($\pm$5.98)	56.51($\pm$10.2)	54.74($\pm$14.5)	76.03($\pm$8.79)	76.19($\pm$7.42)	**83.85**($\pm$9.14)	79.14($\pm$11.3)
$\overline{\text{vol}}_{2,t}$	0.26($\pm$0.08)	**0.08**($\pm$0.03)	0.44($\pm$0.11)	0.22($\pm$0.09)	0.23($\pm$0.04)	0.08($\pm$0.04)	0.15($\pm$0.04)	0.11($\pm$0.03)
$\overline{T}_{2,t}$	0.27($\pm$0.11)	0.15($\pm$0.09)	0.57($\pm$0.13)	0.17($\pm$0.05)	0.07($\pm$0.03)	**0.05**($\pm$0.02)	0.13($\pm$0.02)	0.19($\pm$0.04)
$\overline{E}_{2,t}$	0.48($\pm$0.16)	0.26($\pm$0.11)	0.51($\pm$0.14)	**0.09**($\pm$0.05)	0.72($\pm$0.09)	0.37($\pm$0.09)	0.16($\pm$0.03)	0.12($\pm$0.05)
$\overline{\text{rwd}}_{2,t}$	62.65($\pm$8.63)	68.76($\pm$4.97)	46.73($\pm$16.4)	75.14($\pm$10.1)	76.08($\pm$8.51)	76.58($\pm$6.37)	**77.86**($\pm$8.83)	75.93($\pm$7.88)
$\overline{\text{vol}}_{3,t}$	1.25($\pm$0.12)	0.32($\pm$0.07)	0.64($\pm$0.09)	1.05($\pm$0.13)	0.52($\pm$0.03)	0.25($\pm$0.06)	0.34($\pm$0.07)	**0.18**($\pm$0.05)
$\overline{T}_{3,t}$	0.34($\pm$0.10)	0.16($\pm$0.09)	**0.11**($\pm$0.03)	0.21($\pm$0.11)	0.21($\pm$0.06)	0.25($\pm$0.06)	0.21($\pm$0.05)	0.28($\pm$0.14)
$\overline{E}_{3,t}$	0.16($\pm$0.05)	**0.07**($\pm$0.03)	0.59($\pm$0.12)	0.14($\pm$0.05)	0.32($\pm$0.12)	0.11($\pm$0.05)	0.25($\pm$0.05)	0.34($\pm$0.16)
$\overline{\text{rwd}}_{3,t}$	71.85($\pm$7.62)	77.95($\pm$3.08)	67.28($\pm$10.9)	73.33($\pm$5.81)	77.67($\pm$6.25)	79.63($\pm$4.43)	81.56($\pm$6.51)	**82.75**($\pm$8.22)
$\text{vol}^{\text{total}}$	66.84($\pm$4.06)	27.62($\pm$4.42)	37.38($\pm$6.07)	**8.08**($\pm$4.06)	23.76($\pm$4.93)	10.96($\pm$3.34)	19.22($\pm$3.34)	22.31($\pm$2.16)
T^{total}	47.24($\pm$6.27)	34.75($\pm$7.40)	34.75($\pm$7.40)	9.35($\pm$2.89)	11.9($\pm$3.15)	**3.95**($\pm$2.42)	6.89($\pm$1.33)	4.70($\pm$2.46)
E^{total}	13.26($\pm$3.62)	1.35($\pm$1.35)	20.27($\pm$3.48)	**1.07**($\pm$0.24)	12.32($\pm$2.32)	4.60($\pm$1.36)	10.23($\pm$2.28)	9.91($\pm$2.01)
$\text{rwd}^{\text{total}}$	183.42($\pm$25.92)	212.12($\pm$14.03)	170.52($\pm$37.5)	203.3($\pm$30.41)	229.78($\pm$23.54)	232.4($\pm$18.22)	**245.27**($\pm$24.48)	237.82($\pm$27.4)
HVI	0.0926	0.5154	0.1505	0.3284	0.7045	0.6866	**0.7258**	0.6576

[1] $\overline{\text{vol}}$, $\overline{T}$, $\overline{E}$, $\overline{\text{rwd}}$ denote communication overheads (Mbits), latency (s), energy cost (J) and average reward (no units), respectively. The superscript total represents the sum over all tasks. HVI denotes the hypervolume indicator for multi-objective performance (higher is better).
[2] We conduct five independent repeated experiments with different random seeds and train for 35 communication rounds with three SPs.

global rewards, we adopt the hypervolume indicator (HVI) [50], [51]—a standard metric in multi-objective optimization—to assess the quality of the Pareto front for per-SP rewards. Specifically, let

$$\mathcal{P}_{(\text{alg})} = \left\{ \mathbf{v}^{(m)} = \left(\text{rwd}_1^{(m)}, \text{rwd}_2^{(m)}, \ldots, \text{rwd}_R^{(m)} \right) \right\}_{m=1}^{\text{total runs}} \tag{46}$$

denote the set of reward vectors obtained from different independent training runs by an algorithm, where $\text{rwd}_r^{(m)}$ is the average reward of SP r in the m-th run. Given a reference point $\mathbf{v}^{\text{ref}} \in \mathbb{R}^R$ dominated by all solutions, the HVI is computed as

$$\text{HVI} = \varpi \left(\bigcup_{\mathbf{v} \in \mathcal{P}_{\text{alg}}} [\text{rwd}_1, \text{rwd}_1^{\text{ref}}] \times \cdots [\text{rwd}_R, \text{rwd}_R^{\text{ref}}] \right), \tag{47}$$

where $\varpi(\cdot)$ denotes the Lebesgue measure [50], a larger HVI inidcates both closer alignment with ideal per-SP rewards and a broader spectrum of trade-offs. Additional HVI computation details are given in Appendix B-A, available online.

B. Simulation Results

1) Comparison With Baselines: Fig. 3 depicts the training process of the PAC-based agent across three FL tasks. The reward trajectories converge to stability within several episodes, empirically validating the convergence guarantee established in Theorem 2 and ensuring the trained agents are effectively utilized for subsequent testing. Table IV and Fig. 4 underscore the superior performance of the proposed PAC-MCoFL in balancing model accuracy, communication efficiency, and energy consumption under multi-SP scenarios. As evidenced in Table IV, PAC-MCoFL significantly outperforms non-MARL baselines, with a 10.4% increase in total reward and a 121% improvement in HVI score over FedDQ-h and AdaQuantFL-h, respectively, which are limited to solving a part of the optimization task.

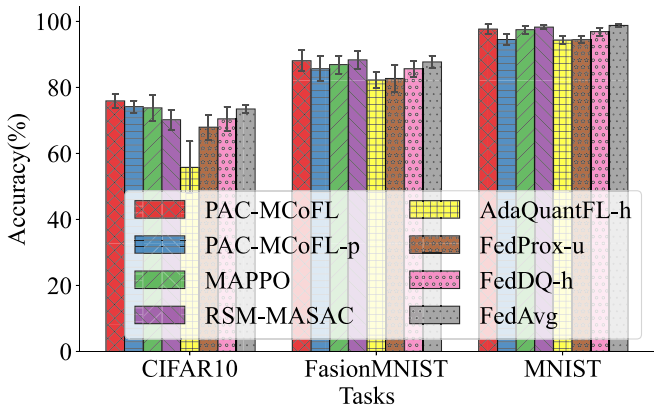

Fig. 4. Average test accuracy for the three services after a fixed number of FL global training rounds.

This superiority thereby underscores the pronounced benefits of holistic joint optimization over localized or heuristic strategies. Within the more competitive landscape of MARL frameworks, it achieves the highest total reward (245.27 ± 24.48) and demonstrates a 6.7% lead over MAPPO and a 5.5% lead over RSM-MASAC. While total reward serves as a valuable macro-indicator, the HVI offers a more nuanced assessment of Pareto efficiency in this multi-objective, competitive setting. PAC-MCoFL again leads with the highest HVI of 0.7258, surpassing MAPPO by 3.0% and RSM-MASAC by 5.7%. This quantitative superiority in both metrics indicates that PAC-MCoFL not only maximizes collective output but also discovers a more efficient and balanced equilibrium. The variant PAC-MCoFL-p affirms its role as a computationally scalable yet potent alternative, as it secures the second-highest total reward (237.82 ± 27.4) and a robust HVI of 0.6576. Meanwhile, the modest, quantifiable performance gap between PAC-MCoFL and PAC-MCoFL-p also practically manifests the "provably bounded error" established in Theorem 1. It explicitly demonstrates that our

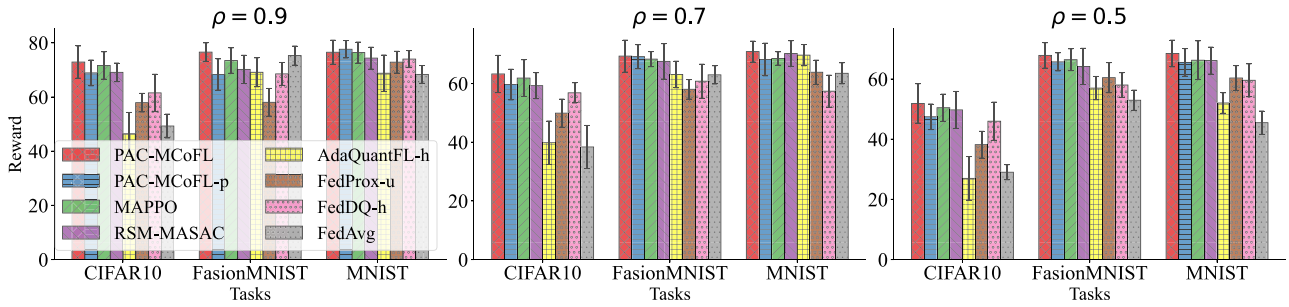

Fig. 5. Comparison of test accuracy across algorithms under varying non-IID degrees. We conduct five independent repeated experiments with different random seeds and train for 35 communication rounds with three SPs.

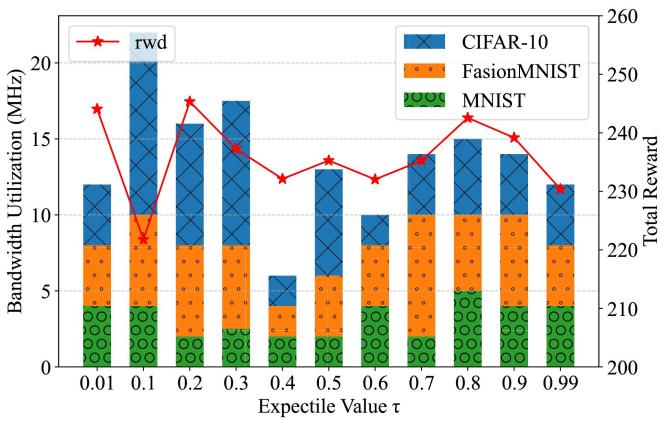

Fig. 6. Per-task bandwidth resource assignment and total reward under different expectile values. Total Reward means the sum of the average reward of all SPs.

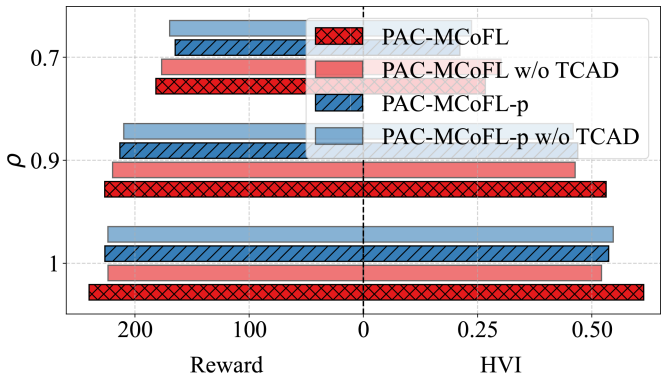

Fig. 7. Impact of the TCAD mechanism.

parameterized generator provides a computationally scalable solution with only a minimal and acceptable deviation from the exhaustive search policy, thereby justifying its use. Furthermore, Fig. 4 shows the average task accuracy achieved by various algorithms. PAC-MCoFL leads with the highest accuracy over all tasks, while PAC-MCoFL-p's advantage is most pronounced in the more complex CIFAR-10 task where its accuracy is only marginally inferior to PAC-MCoFL. In summary, incorporating policy conjecture of other SPs effectively benefits multi-SP FL communication-computation co-optimization.

Fig. 5 illustrates the comparison of test accuracy across different algorithms under varying non-IID degrees. As non-IID severity increases (ρ decreases), both PAC-MCoFL and PAC-MCoFL-p exhibit a graceful performance degradation, yet PAC-MCoFL outperforms the other algorithms in most cases. Even under extreme heterogeneity, i.e. $\rho = 0.5$, PAC-MCoFL consistently outperforms MAPPO and RSM-MASAC baseline, with reward improvements ranging from 1.8–3.1% and 2.1–5.6%, respectively, across different tasks. This robustness highlights PAC-MCoFL's adaptive optimization capabilities in non-IID multi-SP environments.

2) Impact of Expectile Regression: Fig. 6 shows the impact of the expectile value τ on SPs' bandwidth allocation and the total system reward. Through an asymmetric penalty mechanism, τ implicitly guides SP policies, resulting in a U-shaped relationship between τ and bandwidth utilization. When $\tau < 0.5$, the critic network imposes a heavier penalty on negative TD errors, promoting aggressive policies to compensate for underestimated Q-values. This leads to bandwidth over-allocation and a higher proportion of resources for the complex CIFAR-10 task. Conversely, $\tau > 0.5$ yields more conservative and balanced bandwidth allocation. While the overall system reward fluctuates within the range of 230–240, both extremes (i.e., $\tau = 0.1$ and $\tau = 0.99$) can cause up to 10.4% performance degradation. Therefore, we set $\tau = 0.5$ hereafter for simplicity.

3) Impact of TCAD Mechanism: To validate the efficacy of the proposed TCAD, we conducted an ablation study by replacing it with a naive discretization variant.[5] As illustrated in Fig. 7, removing TCAD consistently results in a degradation in both total reward and the HVI score for the PAC-MCoFL and PAC-MCoFL-p frameworks alike. Across all scenarios, TCAD delivers an average increase of approximately 7.6% in total reward and 14.2% in HVI compared to the w/o TCAD variant. This performance gap underscores the limitations of this naive variant, which restricts the agent to coarse, disjointed action selections. In contrast, TCAD's capacity for fine-grained, incremental control is crucial for discovering sophisticated policies and navigating complex action spaces effectively.

[5]See Appendix C-B, available online for related configurations and further discussion.

TABLE V
GPU MEMORY CONSUMPTION (IN MB) FOR DIFFERENT METHODS AND VALUES OF R

Method	R=2	R=3	R=4	R=5
MAPPO	512M	498M	524M	524M
PAC-MCoFL	536M	2132M	OOM[1]	OOM
PAC-MCoFL-p	496M	490M	508M	508M

[1] Out of Memory (OOM) denotes that the memory requirements exceeded the capacity of the current experimental hardware (NVIDIA GeForce RTX 3090).

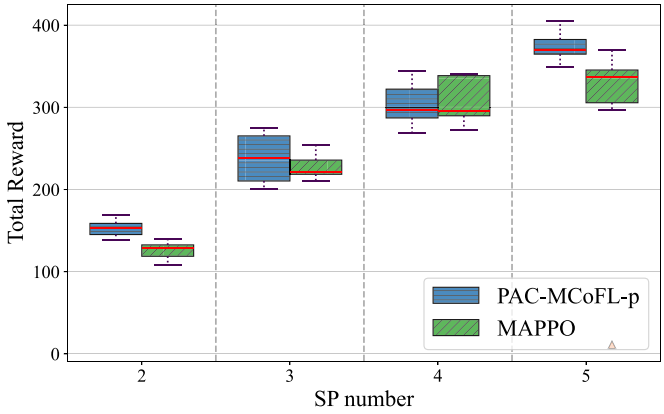

Fig. 8. Experiment with broader multi-agent complexities. The task datasets and corresponding models used by different SPs under different R will be given in detail in the Appendix B-B, available online. Total Reward is the sum of the average rewards across all SPs.

4) Impact of SP Numbers: The exhaustive policy conjecture in (23) incurs a computational complexity that grows exponentially with the number of SPs, rendering PAC-MCoFL intractable in larger scenarios. PAC-MCOFL-p is specifically designed to overcome this scalability bottleneck. Table V shows that the standard PAC-MCoFL encounters out-of-memory (OOM) errors for $R \geq 4$ on our hardware, whereas PAC-MCOFL-p remains computationally feasible. Fig. 8 further reveals that this efficiency does not compromise performance. PAC-MCOFL-p consistently outperforms the scalable baseline MAPPO,[6] achieving an average reward improvement of approximately 21.3%. Crucially, this scalability analysis also accounts for increased task heterogeneity, as each new SP introduces a task with distinct characteristics as detailed in Appendix B-B, available online.

5) Impact of Different Weight Factors $\boldsymbol{\sigma}$: To evaluate the impact of the different terms introduced in the objective function, specifically the reward function in (17), on the performance of PAC-MCoFL, we sequentially set σ_1 to σ_4 to zero in turn. We then train the agents accordingly, and test the performance across various metrics within a fixed FL global training round. The results are summarized as shown in Table VI, where accuracy $\Gamma_{r,t}^{\max}$ refers to the obtained maximum value, while the other metrics are averaged over the FL rounds. It can be observed that the absence of the accuracy factor in the reward function ($\sigma_1 = 0$) leads to the most significant performance degradation, with PAC-MCoFL reward dropping by 41.6%, 14.8%, and 4.7% across the three tasks, underscoring the critical importance of accuracy. Excluding the adversarial factor ($\sigma_2 = 0$) has minimal impact on the overall reward, though it negatively affects individual task metrics such as $T_{r,t}$, $E_{r,t}$, and $\mathrm{vol}_{r,t}$. Omitting latency ($\sigma_3 = 0$) reduces CIFAR-10 and Fashion-MNIST reward by 35.13% and 34.3%, respectively, while excluding energy consumption ($\sigma_4 = 0$) results in a 32.6% drop in the CIFAR-10 reward. These results suggest that without latency and energy constraints, the adversarial factor becomes overly dominant, leading to weaker performance in certain tasks. The weight factors study reveals that accuracy is the most critical factor for PAC-MCoFL, with latency and energy constraints also playing key roles in maintaining task balance. While removing the adversarial factor has a minimal overall impact, it is likely to degrade individual task performance.

6) Different Number of Clients At Different non-IID Degrees: We investigate the impact of different degrees of non-IID settings and varying numbers of clients on the performance of PAC-MCoFL. The results in Table VII reveal two primary trends. First, scaling the number of clients from 5 to 15 introduces a clear tension between resource overhead and model performance. As expected, total energy E^{total}, delay T^{total}, and communication overheads $\mathrm{vol}^{\mathrm{total}}$ exhibit a general upward trend with an increase of N, reflecting the costs of coordinating a larger client pool. Concurrently, a consistent degradation in final test accuracy is observed across all tasks, highlighting the inherent challenges of scaling. Second, for a fixed client number, increasing data heterogeneity (i.e., decreasing ρ) not only markedly reduces final model accuracy but also simultaneously inflates the resource overhead required for training. This demonstrates that the framework adaptively expends more resources to counteract the negative effects of statistical heterogeneity.

7) Effect of Different Jitter Variance Factors: In PAC-MCoFL, decision-making occurs at the SP level, i.e., on the server side. As mentioned in Section IV-A, the behavior of clients, such as quantization level $q_{i,r,t}$ and CPU frequency $f_{i,r,t}$, are sampled based on the SP's decisions, with an additional variance Σ_q and Σ_f. To further assess performance, we evaluate different values of Σ, conducting five independent experiments for each configuration. Figs. 9 and 10 depict the reward achieved by the three FL tasks during a fixed number of global training rounds under varying Σ_q and Σ_f settings. As shown in Fig. 9, increasing Σ_q results in lower reward and greater variability. In particular, when $\Sigma_q = 3$, the reward significantly deteriorates across all three tasks. In the CIFAR-10 task, $\Sigma_q = 2.5$ and $\Sigma_q = 3$ even exhibit a downward trend, indicating that larger Σ_q values may adversely affect the system's convergence performance. This suggests that when quantization strategies among FL clients vary too widely, the resulting divergence in the uploaded local models can hinder convergence. Similarly, Fig. 10 shows that $\Sigma_f = 2$ and $\Sigma_f = 2.5$ lead to the poorest performance across all tasks, with $\Sigma_f = 2.5$ even exhibiting a lack of convergence in reward for the MNIST task. In contrast, the results for $\Sigma_f = 0.5$ and $\Sigma_f = 1$ exhibit only minor fluctuations compared to the BASELINE, indicating that small Σ_f values have a negligible impact on system performance.

[6] Notably, MAPPO does not involve policy conjecture, thereby avoiding the costly enumeration of all possible joint actions of neighboring agents.

TABLE VI
RESULTS OF DIFFERENT OBJECTIVE WEIGHT FACTORS $\boldsymbol{\sigma}$

Service	*CIFAR-10*					*Fashion-MNIST*					*MNIST*				
Metric	$\Gamma_{1,t}^{\max}$	$\overline{\text{vol}}_{1,t}$	$\overline{T}_{1,t}$	$\overline{E}_{1,t}$	$\overline{\text{rwd}}$	$\Gamma_{2,t}^{\max}$	$\overline{\text{vol}}$	$\overline{T}_{2,t}$	$\overline{E}_{2,t}$	$\overline{\text{rwd}}$	$\Gamma_{3,t}^{\max}$	$\overline{\text{vol}}$	$\overline{T}_{3,t}$	$\overline{E}_{3,t}$	$\overline{\text{rwd}}$
$\sigma_{1\sim4}\neq 0$	**78.55%**	18.74	**6.62**	**9.96**	**83.85**	**91.01%**	0.146	0.132	**0.161**	**77.86**	**98.23%**	0.315	**0.162**	0.21	81.56
$\sigma_1=0$	40.52%	~~13.67~~	~~5.41~~	~~6.58~~	~~48.92~~	88.64%	0.123	**0.08**	0.21	64.76	97.84%	0.35	0.19	0.61	77.69
$\sigma_2=0$	75.28%	**16.7**	6.98	13.25	76.03	86.02%	**0.092**	0.152	0.178	62.14	96.07%	**0.27**	0.24	0.353	**82.82**
$\sigma_3=0$	68.13%	18.14	7.42	11.53	54.39	59.35%	~~0.031~~	~~0.121~~	~~0.082~~	~~49.93~~	97.68%	0.46	0.232	**0.271**	71.85
$\sigma_4=0$	46.06%	~~22.13~~	~~23.17~~	~~7.44~~	~~56.51~~	88.32%	0.224	0.242	0.85	72.82	97.78%	0.68	0.315	0.518	67.28

[1] $\overline{\text{vol}}_{r,t}, \overline{T}_{r,t}, \overline{E}_{r,t}$ which are in Mbits, s, and J, respectively, represent the average communication overheads, latency, and energy consumption consumed during FL training rounds. $\overline{\text{rwd}}$ is the average reward and has no units.
[2] Data with a strikethrough indicates that the business model does not eventually converge (accuracy still fluctuates significantly), and bolded data indicates an advantage in comparison with the same metric.

TABLE VII
PERFORMANCE COMPARISON UNDER DIFFERENT NUMBERS OF CLIENTS (N) AND DEGREES OF NON-IID LEVEL (ρ)

Clients (N)	**Non-IID** (ρ)	**Resource Overheads**			**Final Test Accuracy (%)**		
		E^{total} (J)	T^{total} (s)	$\text{vol}^{\text{total}}$ (Mbits)	**CIFAR-10**	**Fashion-MNIST**	**MNIST**
5	1.0	360.34(±38.02)	205.74(±36.40)	543.11(±58.26)	75.89(±2.54)	88.06(±3.24)	97.65(±1.25)
	0.9	374.22(±32.65)	351.96(±47.63)	634.74(±57.64)	66.08(±6.94)	83.64(±3.77)	91.56(±1.12)
	0.7	324.96(±28.96)	281.39(±32.50)	867.19(±80.24)	55.26(±6.34)	76.35(±3.44)	86.89(±2.57)
	0.5	521.92(±40.25)	384.33(±43.56)	1156.61(±108.68)	45.89(±7.64)	69.86(±4.21)	73.50(±2.42)
10	1.0	488.79(±48.99)	286.56(±21.26)	991.21(±90.87)	74.76(±3.12)	86.84(±3.21)	97.67(±1.21)
	0.9	633.69(±64.68)	369.98(±40.38)	1416.21(±118.68)	68.76(±7.24)	80.64(±4.20)	94.17(±1.50)
	0.7	760.75(±65.96)	426.75(±44.50)	2010.98(±155.48)	62.98(±6.69)	72.38(±4.23)	91.43(±3.48)
	0.5	788.81(±88.48)	325.03(±51.11)	1923.45(±143.68)	48.76(±8.03)	69.75(±5.54)	87.24(±4.23)
15	1.0	924.45(±53.79)	392.63(±24.86)	1379.64(±88.37)	74.04(±2.32)	83.17(±3.17)	94.16(±1.46)
	0.9	876.56(±58.63)	394.98(±39.30)	1250.35(±118.68)	68.84(±3.19)	80.22(±2.88)	93.85(±1.29)
	0.7	986.68(±42.36)	467.65(±46.96)	2656.65(±160.69)	67.26(±5.26)	78.79(±2.16)	89.77(±4.26)
	0.5	906.68(±46.63)	405.05(±36.12)	2170.15(±152.44)	61.32(±5.21)	75.67(±4.55)	86.58(±2.16)

[*] $\text{vol}^{\text{total}}$, T^{total}, E^{total} denote the sum of communication overheads, latency and energy cost over all tasks, respectively. We conduct five independent repeated experiments with different random seeds and train for 35 communication rounds with three SPs.

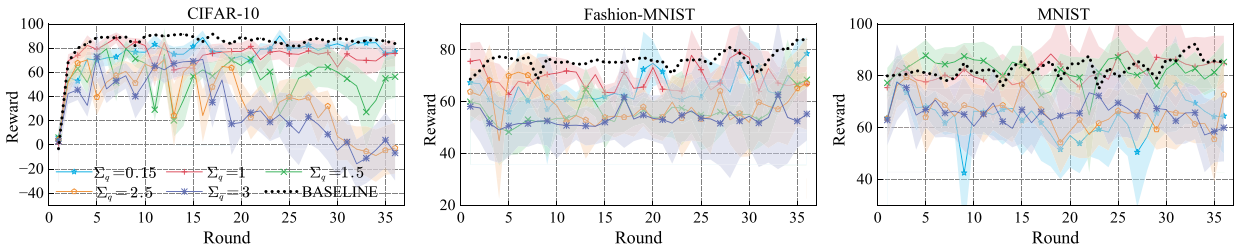

Fig. 9. Impact of different coefficient Σ_q on system performance. The "BASELINE" configuration corresponds to the simulation parameters in Table III (i.e., $\Sigma_q = 0.5$ and $\Sigma_f = 0.25$).

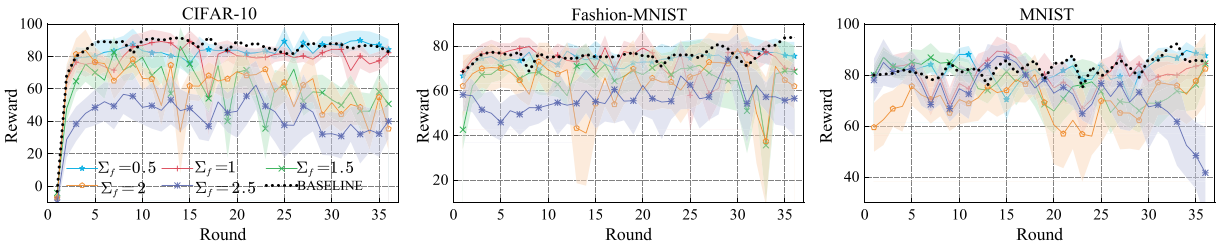

Fig. 10. Impact of different coefficient Σ_f on system performance. The "BASELINE" configuration corresponds to the simulation parameters in Table III (i.e., $\Sigma_q = 0.5$ and $\Sigma_f = 0.25$).

VI. Conclusion and Future Work

In this paper, we have developed the PAC-MCoFL—a novel game-theoretic MARL framework for jointly optimizing communication and computation in non-cooperative multi-SP FL. By integrating PAC principles with expectile regression, PAC-MCoFL steers various SP agents toward Pareto-optimal equilibria while capturing their heterogeneous risk preferences. Furthermore, PAC-MCoFL leverages a TCAD mechanism to enable fine-grained, multi-dimensional action control. To ensure both practicality and scalability, we have also designed a lightweight variant, PAC-MCoFL-p, which provably overcomes the exponential complexity of policy conjecture under bounded-error guarantees. In addition to the theoretical convergence results, extensive simulations demonstrate superior performance. Prominently, PAC-MCoFL achieves improvements of approximately 5.8% in total reward and 4.2% in HVI over strong MARL-based baselines. Subsequent research will investigate adaptation mechanisms, such as dynamic τ-adaptation for time-varying risk profiles and automatic tuning of the reward weights σ to balance multi-objective trade-offs, as well as the extension to mixed cooperative-competitive settings.

References

[1] Z. Zhao et al., "AQUILA: Communication efficient federated learning with adaptive quantization in device selection strategy," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 7363–7376, Jun. 2024.

[2] "Private cloud compute: A new frontier for ai privacy in the cloud," 2024. [Online]. Available: https://security.apple.com/blog/private-cloud-compute/

[3] Z. Zhang, Z. Gao, Y. Guo, and Y. Gong, "Scalable and low-latency federated learning with cooperative mobile edge networking," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 812–822, Jan. 2024.

[4] 3GPP, "5G; System architecture for the 5G System (5GS)," 3rd Generation Partnership Project (3GPP), Tech. Rep. TS 23.501 (Release 16), Oct. 2020.

[5] M. Kim, W. Saad, M. Mozaffari, and M. Debbah, "Green, quantized federated learning over wireless networks: An energy-efficient design," *IEEE Trans. Wireless Commun.*, vol. 23, no. 2, pp. 1386–1402, Feb. 2024.

[6] D. Kwon, J. Jeon, S. Park, J. Kim, and S. Cho, "Multiagent DDPG-based deep learning for smart ocean federated learning IoT networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9895–9903, Oct. 2020.

[7] S. Q. Zhang, J. Lin, and Q. Zhang, "A multi-agent reinforcement learning approach for efficient client selection in federated learning," in *Proc. AAAI Conf. Artif. Intell.*, Vancouver, Canada, 2022, pp. 9091–9099.

[8] S. Zheng, Y. Dong, and X. Chen, "Deep reinforcement learning-based quantization for federated learning," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Glasgow, U.K., 2023, pp. 1–6.

[9] W. Mao, X. Lu, Y. Jiang, and H. Zheng, "Joint client selection and bandwidth allocation of wireless federated learning by deep reinforcement learning," *IEEE Trans. Serv. Comput.*, vol. 17, no. 1, pp. 336–348, Jan./Feb. 2024.

[10] K. Park, M. Kim, and L. Park, "NeuSaver: Neural adaptive power consumption optimization for mobile video streaming," *IEEE Trans. Mobile Comput.*, vol. 22, no. 11, pp. 6633–6646, Nov. 2023.

[11] E. Amiri, N. Wang, M. Shojafar, and R. Tafazolli, "Edge-AI empowered dynamic VNF splitting in O-RAN slicing: A federated DRL approach," *IEEE Commun. Lett.*, vol. 28, no. 2, pp. 318–322, Feb. 2024.

[12] J. Xu, H. Wang, and L. Chen, "Bandwidth allocation for multiple federated learning services in wireless edge networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2534–2546, Apr. 2022.

[13] C. Yu et al., "The surprising effectiveness of PPO in cooperative multi-agent games," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, New Orleans, USA, 2021, pp. 24611–24624.

[14] Y. Yang et al., "Mean field multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2018, pp. 5571–5580.

[15] E. Pei, Y. Huang, L. Zhang, Y. Li, and J. Zhang, "Intelligent access to unlicensed spectrum: A mean field based deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 4, pp. 2325–2337, Apr. 2023.

[16] B. G. Le and V. C. Ta, "Toward finding strong Pareto optimal policies in multi-agent reinforcement learning," *Mach. Learn.*, vol. 114, no. 3, pp. 1–20, Feb. 2025.

[17] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit Q-learning," in *Proc. Int. Conf. Learn. Representations*, 2022.

[18] G. Papoudakis et al., "Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Beijing, China, 2021, pp. 15–19.

[19] F. Christianos, G. Papoudakis, and S. V. Albrecht, "Pareto actor-critic for equilibrium selection in multi-agent reinforcement learning," *Trans. Mach. Learn. Res.*, Oct. 2023, issn: 2835–8856. [Online]. Available: https://openreview.net/forum?id=3AzqYa18ah

[20] Y. Liu, X. Zhang, Z. Zeng, and S. Yu, "FedEco: Achieving energy-efficient federated learning by hyperparameter adaptive tuning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 10, no. 6, pp. 2311–2324, Dec. 2024.

[21] J. Xue, Z. Qu, J. Xu, Y. Liu, and Z. Lu, "Bandwidth allocation for federated learning with wireless providers and cost constraints," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 7470–7482, Jun. 2024.

[22] M. N. H. Nguyen, N. H. Tran, Y. K. Tun, Z. Han, and C. S. Hong, "Toward multiple federated learning services resource sharing in mobile edge networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 541–555, Jan. 2023.

[23] Y. Bai et al., "Multicore federated learning for mobile-edge computing platforms," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5940–5952, Apr. 2023.

[24] D. Jhunjhunwala et al., "Adaptive quantization of model updates for communication-efficient federated learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 3110–3114.

[25] Z. Mo et al., "FedDQ: A communication-efficient federated learning approach for Internet of Vehicles," *J. Syst. Archit.*, vol. 131, Oct. 2022, Art. no. 102690.

[26] D. Wen, K.-J. Jeon, and K. Huang, "Federated dropout—a simple approach for enabling federated learning on resource constrained devices," *IEEE Wireless Commun. Lett.*, vol. 11, no. 5, pp. 923–927, May 2022.

[27] K. Moghaddasi, S. Rajabi, and F. S. Gharehchopogh, "An enhanced asynchronous advantage actor-critic-based algorithm for performance optimization in mobile edge computing-enabled Internet of Vehicles networks," *Peer-to-Peer Netw. Appl.*, vol. 17, no. 3, pp. 1169–1189, May 2024.

[28] K. Moghaddasi et al., "An energy-efficient data offloading strategy for 5G-enabled vehicular edge computing networks using double deep q-network," *Wireless Pers. Commun.*, vol. 133, no. 3, pp. 2019–2064, Dec. 2023.

[29] A. M. Rahmani et al., "Self-learning adaptive power management scheme for energy-efficient IoT-mec systems using soft actor-critic algorithm," *Internet Things*, vol. 31, May 2025, Art. no. 101587.

[30] K. Moghaddasi, S. Rajabi, and F. S. Gharehchopogh, "Multi-objective secure task offloading strategy for blockchain-enabled IoV-MEC systems: A double deep Q-network approach," *IEEE Access*, vol. 12, pp. 3437–3463, 2024.

[31] S. Martello and P. Toth, "A bound and bound algorithm for the zero-one multiple knapsack problem," *Discret. Appl. Math.*, vol. 3, pp. 275–288, Nov. 1981.

[32] Y. Yu, D. Chen, X. Tang, T. Song, C. S. Hong, and Z. Han, "Incentive framework for cross-device federated learning and analytics with multiple tasks based on a multi-leader-follower game," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3749–3761, Sep./Oct. 2022.

[33] S. Fu, F. Dong, D. Shen, J. Zhang, Z. Huang, and Q. He, "Joint optimization of device selection and resource allocation for multiple federations in federated edge learning," *IEEE Trans. Serv. Comput.*, vol. 17, no. 1, pp. 251–262, Jan./Feb. 2024.

[34] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *J. Mach. Learn. Res.*, vol. 4, no. 11, pp. 1039–1069, Dec. 2003.

[35] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proc. AAAI Conf. Artif. Intell.*, Madison, Wisconsin, USA, 1998, pp. 746–752.

[36] J. N. Foerster et al., "Counterfactual multi-agent policy gradients," in *Proc. AAAI Conf. Artif. Intell.*, New Orleans, Louisiana, USA, Feb. 2018.

[37] J. Wang et al., "SHAQ: Incorporating shapley value theory into multi-agent Q-learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, New Orleans, LA, USA, Dec. 2022, pp. 5941–5954.
[38] H. B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, 2017, pp. 1273–1282.
[39] D. Alistarh et al., "QSGD: Randomized quantization for communication-optimal stochastic gradient descent," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Long Beach, California, USA, 2017, pp. 1707–1718.
[40] T. Yu et al., "Proximal policy optimization-based federated client selection for Internet of Vehicles," in *Proc. IEEE/CIC Int. Conf. Commun. China*, Sanshui, Foshan, China, 2022, pp. 648–653.
[41] 3GPP, "3rd generation partnership project; technical specification group core network and terminals; 5G system; network exposure function northbound apis; stage 3 (release 18)," 3GPP, Tech. Rep. TR 25.522 V18.6.0 (2024-06), 2024.
[42] M. J. Guyer and A. Rapoport, "2 × 2 games played once," *J. Conflict Resolution*, vol. 16, no. 3, pp. 409–431, Sep. 1972.
[43] J. Zhao et al., "Actor-critic for multi-agent reinforcement learning with self-attention," *Int. J. Pattern Recognit Artif Intell.*, vol. 36, no. 09, Jun. 2022, Art. no. 2252014.
[44] T. Yu, X. Wang, J. Hu, and J. Yang, "Multi-agent proximal policy optimization-based dynamic client selection for federated AI in 6G-oriented Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13611–13624, Sep. 2024.
[45] T. Jaakkola, M. I. Jordan, and S. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Comput.*, vol. 6, pp. 1185–1201, Nov. 1994.
[46] C. Szepesvári and M. L. Littman, "A unified analysis of value-function-based reinforcement-learning algorithms," *Neural Comput.*, vol. 11, no. 8, pp. 2017–2060, Nov. 1999.
[47] Z. Jiang et al., "Computation and communication efficient federated learning with adaptive model pruning," *IEEE Trans. Mobile Comput.*, vol. 23, no. 3, pp. 2003–2021, Mar. 2024.
[48] T. Li et al., "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 429–450.
[49] X. Yu, R. Li, C. Liang, and Z. Zhao, "Communication-efficient soft actor–critic policy collaboration via regulated segment mixture," *IEEE Internet Things J.*, vol. 12, no. 4, pp. 3929–3947, Feb. 2025.
[50] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011.
[51] A. P. Guerreiro, C. M. Fonseca, and L. Paquete, "The hypervolume indicator: Computational problems and algorithms," *ACM Comput. Surv.*, vol. 54, no. 6, Jul. 2021.

Renxuan Tan (Student Member, IEEE) received the BE degree in communication engineering from Sichuan University, Chengdu, China, in 2024. He is currently working toward the PhD degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. His research interests include reinforcement learning, wireless communications.

Rongpeng Li (Senior Member, IEEE) received the BE degree from Xidian University, Xi'an, China, in 2010, and the PhD degree from Zhejiang University, Hangzhou, China, in 2015. He is currently an associate professor with the College of Information Science and Electronic Engineering, Zhejiang University. From August 2015 to September 2016, he was a research engineer with the Wireless Communication Laboratory, Huawei Technologies Company Ltd., Shanghai, China. He was a Visiting Scholar with the Department of Computer Science and Technology, University of Cambridge, Cambridge, U.K., from February 2020 to August 2020. His current research interests focus on networked intelligence for comprehensive efficiency (NICE).

Xiaoxue Yu (Student Member, IEEE) received the BE degree in communication engineering from Xidian University, Xi'an, China. She is currently working toward the PhD degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. Her research interests include communications in distributed learning and multiagent reinforcement learning.

Xianfu Chen (Senior Member, IEEE) received the PhD degree (with Hons.) from the Zhejiang University, Hangzhou, China, in 2012. In 2012, he joined the VTT Technical Research Centre of Finland, Oulu, Finland, as a research scientist and as a senior scientist from 2013 to 2023. He is currently a chief research engineer with the Shenzhen CyberAray Network Technology Co., Ltd, Shenzhen, China. His research interests include various aspects of wireless communications and networking, with emphasis on human-level and artificial intelligence for resource awareness in next-generation communication networks. He was the recipient of the 2021 IEEE Communications Society Outstanding Paper Award, and the 2021 *IEEE Internet of Things Journal Best Paper Award*. He is an editor of *IEEE Open Journal of the Communications Society*, an academic editor of *Wireless Communications and Mobile Computing*, and an associate editor of China Communications.

Xing Xu received the PhD degree in the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. He is now a data engineer of Information and Communication Branch of State Grid Hebei Electric Power Co., Ltd., Shijiazhuang, China. His research interests include collective intelligence, reinforcement learning, multi-agent reinforcement learning, federated learning, and data mining.

Zhifeng Zhao (Senior Member, IEEE) received the BE degree in computer science, the ME degree in communication and information systems, and the PhD degree in communication and information systems from the PLA University of Science and Technology, Nanjing, China, in 1996, 1999, and 2002, respectively. From 2002 to 2004, he acted as a post-doctoral researcher with Zhejiang University, Hangzhou, China, where his researches were focused on multimedia next-generation networks (NGNs) and softswitch technology for energy efficiency. Currently, he is with the Zhejiang Lab, Hangzhou as the chief engineering Officer. His research areas include software defined networks (SDNs), wireless network in 6G, computing networks, and collective intelligence. He is the Symposium co-chair of ChinaCom 2009 and 2010. He is the Technical Program Committee (TPC) co-chair of the 10th IEEE International Symposium on Communication and Information Technology (ISCIT 2010).