Federated Multi-Agent Deep Reinforcement Learning (Fed-MADRL) for Dynamic Spectrum Access

Hao-Hsuan Chang, Yifei Song, Thinh T. Doan, and Lingjia Liu

***Abstract*—Dynamic spectrum access (DSA) has been introduced as a promising technology that allows a secondary system to access the licensed spectrum of the primary system to improve spectrum utilization. In this paper, we introduce Fed-MADRL by incorporating federated learning (FL) and multi-agent deep reinforcement learning (MADRL) to design a collaborative DSA strategy. Our Fed-MADRL scheme employs FL to enable multiple users to collaboratively optimize the system goal without sharing their training data. By keeping all the training data at the user end, FL improves the communication efficiency and strengthens user data privacy. To further reduce the communication overheads, each user only shares quantized information. We provide the convergence analysis to characterize the trade-off between the communication efficiency and the system performance. In particular, we show that the introduced method converges at a rate $O(1/K^{1/4})$, where K is the number of FL iterations. To the best of our knowledge, Fed-MADRL is the first work that utilizes FL in DSA networks under quantized communication. Performance evaluation results show that the introduced Fed-MADRL method outperforms the independent learning method and achieves comparable performance with the centralized MADRL method, which requires much higher communication overheads.**

***Index Terms*—Federated learning (FL), Multi-agent deep reinforcement learning (MADRL), Dynamic spectrum access (DSA), Citizens Broadband Radio Service (CBRS), 5G Beyond and 6G.**

I. Introduction

With the development of new wireless technologies and applications, the demand for wireless access has increased remarkably in recent years. According to Cisco's annual internet report, the number of wireless devices is expected to grow at a compound annual growth rate (CAGR) of 10% between 2018 and 2023, reaching 29.3 billion wireless devices by 2023 [1]. To cope with this unprecedented high demand for wireless connections, extending the radio spectrum for commercial use is critical for the fifth-generation (5G) mobile broadband networks. However, the online table of frequency allocation published by the Federal Communications Commission (FCC) demonstrates the extremely congested frequency allocations. Such a crowded frequency allocation makes obtaining new licensed spectrum bands for developing wireless applications costly and challenging. Although the radio spectrum is a precious resource, experimental tests, and investigations from both academia and industries show that the spectrum utilization ratio is actually very low in certain areas [2], [3]. This paradox is caused by the traditional static spectrum management policy that allocates a fixed spectrum band to a single system for exclusive use. The spectrum is underutilized because unlicensed users cannot operate on the licensed spectrum even when licensed users are idle. Therefore, dynamic spectrum access (DSA) has emerged as a promising technique by adopting a hierarchical spectrum access structure with primary users (PUs) and secondary users (SUs) [4]. To be specific, SUs are allowed to access the licensed spectrum when PUs are idle or receive little interference from SUs. In this way, the spectrum can be utilized more efficiently, and thus the spectrum utilization ratio can be increased.

Most model-dependent DSA methods translate the operations in wireless networks into tractable mathematical models to find solutions via conventional optimization methods [5]–[7]. However, with the proliferation of wireless applications in 5G networks, the underlying management of spectrum resources becomes more complicated. Finding a closed-form solution via conventional optimization methods in such complex wireless systems becomes extremely challenging if not impossible, and thus model-dependent DSA methods cannot be can be deployed in real-world 5G systems. To make matters worse, even if a comprehensive and tractable mathematical formulation could be derived, the high computational complexity would render such approach impractical. Therefore, machine learning (ML) approaches have been adopted for the DSA problem since they can manage the spectrum resources dynamically to adapt to an unknown wireless environment.

Model-free reinforcement learning (RL) stands out from many ML-based DSA approaches because it can directly learn the spectrum access policy through interactions with the unknown DSA network [8]. Since there are multiple SUs sharing the DSA environment, the DSA problem naturally falls into the multi-agent reinforcement learning (MARL) setting. To be specific, MARL involves multiple agents interacting with each other in a shared environment, and the local environments of these agents are correlated. Since the performance of one agent's policy is influenced by other agents' policies, the

The authors are with Wireless@Virginia Tech, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. The work of H. Chang, Y. Song and L. Liu was supported in part by US National Science Foundtiona (NSF) under grants ECCS-1811497 and CNS-1811720. The work of Thinh T. Doan was supported in part by the Commonwealth Cyber Initiative, an investment in the advancement of cyber R&D, innovation, and workforce development. The corresponding author is L. Liu (ljliu@ieee.org).

collaborative learning usually has a better performance than that of the independent learning. The collaborative learning is especially important to resource allocation problems in wireless communications because most wireless resources are very limited and the number of competing users is large, which makes the local environments highly correlated. A straightforward implementation of the collaborative learning is the centralized learning, which requires a centralized server to collect the training data from all users to perform the training. Therefore, recent MARL-based solutions for wireless resource allocation problems utilize the centralized training scheme, including DSA [9]–[12], power control [13], [14], and interference coordination [15]. However, the training data are usually distributed among wireless users in different locations, and thus users have to send their collected data to the centralized server for training. The frequent data exchange between a centralized server and wireless users results in high communication overheads in network operations. Furthermore, the user data sent to the server may contain sensitive information such as user location, which may be eavesdropped by hackers causing privacy issues. In the DSA problem, a user's location privacy information may leak because an attacker can infer the location of a user from its spectrum utilization data [16], [17]. Therefore, the idea of the distributed learning has been introduced where users are not required to send their private data to the centralized server. For example, in the entirely distributed learning scheme called independent learning, each wireless user only optimizes its own training task without exchanging information with each other [18]–[21]. Although independent learning can maintain good efficiency and scalability when handling a large number of users, the system performance and the convergence of independent learning are much worse than the centralized training due to a lack of cooperation among the users.

A distributed learning framework called federated learning (FL) is introduced to enable the collaborative learning without collecting the training data in a centralized fashion [22]. In the FL framework, multiple users collaboratively train a shared ML model by only exchanging ML model parameters without sharing their training data. In this way, all the training data will remain at local devices ensuring data privacy. Meanwhile, the communication efficiency can also be improved since only model updates are exchanged between the centralized server and local devices. Lastly, FL will enable cooperation among user devices via the adequate model aggregation process at the centralized server. In this work, we aim to solve the MARL problem using the standard FL framework. Although the FL framework was presented in the areas of wireless communications, most of them focus on solving supervised learning problems [23]–[25]. Studying the effect of FL on MARL problems are more challenging because all agents have to coordinate their actions to find the optimal policy. The FL framework has been applied to some MARL problems in wireless communications [26], [27], but there is lack of theoretical analysis that studies the performance of FL on solving a MARL problem. Furthermore, compared to most MARL algorithms that assume each agent knows the joint state or the joint reward information [28]–[30], our work considers a more general MARL setting where each agent's state is a subset of the joint state, and each agent only receives a local reward. The reason is that the cost of sharing information among multiple agents in the wireless system is high due to the extremely limited wireless communication capacity, and thus it is impractical to assume that each DSA agent can obtain the aggregated reward in every local training iteration. In this work, we provide the convergence analysis of solving this general MARL problem using the FL framework. Meanwhile, motivated by the success of deep reinforcement learning (DRL) in handling large state-action space in most real-world environments, we utilize DRL in the implementation to address the continuous state space in our considered DSA problem. By incorporating FL and multi-agent DRL, we introduce Fed-MADRL, a distributed DSA strategy for SUs to jointly learn a spectrum access policy.

Due to limited bandwidth in wireless systems, we include the model quantization to reduce the communication overheads. To be specific, each user only shares the quantized information with the centralized server, which is a quantized version of its locally updated model by mapping the weights from continuous values to discrete values. The reduced communication overheads may come at the cost of degraded system performance because the shared quantized model is not accurate. To illustrate the cost of using the quantized information for communication, we analyze the effect of the model quantization in our convergence analysis to demonstrate the trade-off between the communication efficiency and the system performance. Existing FL works with quantization assume "zoom in" and "zoom out" quantization, where each agent has to send both the quantization levels and the minimal/maximal values of the quantization range to the centralized server [31]–[33]. It is important to note that the minimal and maximal value of the quantization range are real numbers. Therefore, this quantization method is not realistic because it still requires agents to send real numbers to the centralized server in each FL iteration. If agents are allowed to send real numbers, then the convergence rate is the same as without quantization. On the other hand, we utilize a practical quantization method that only allows users to send quantized values to the centralized server. Using the conventional FL algorithm cannot guarantee the convergence under this quantization method. Therefore, we introduce another step size in the global update to guarantee the FL convergence under this practical quantization scheme.

In this paper, we study the spectrum access strategy for SUs in the Citizens Broadband Radio Service (CBRS) system, which utilizes the 3.5 GHz (3550-3700 MHz) band released by the FCC for shared spectrum usage of federal and commercial users [34]. To be specific, the CBRS system has been opened for spectrum sharing across three tiers of users: Incumbent Users (IUs), Priority Access License (PAL) users, and General Authorized Access (GAA) users. IUs include federal users such as military radars and satellite ground stations, which are the highest tiers and should be protected from possible interference from the lower tiers such as PAL or GAA users. The second tier PAL users are commercial users that are protected from the interference caused by GAA users. PALs are licensed through on spectrum auction, and each PAL

consists of a 10 MHz channel for a 10-year term. Finally, the lowest tier GAA users must not cause harmful interference to IUs or PAL users and should tolerate the interference from them. According to the FCC, an automated frequency coordinator called Spectrum Access System (SAS) will manage the operations among users from the above mentioned three tiers through the assistance from an Environmental Sensing Capability (ESC) which is a sensor network that detects the transmissions of IUs. Therefore, although GAA users do not need to purchase the license for using the CBRS spectrum, they are required to register with the SAS. GAA users have to send spectrum access requests to the SAS before accessing the CBRS band. In this work, we focus on designing the spectrum access algorithm for GAA users.

The main contributions of this work are as follows: 1) We introduced a novel Fed-MADRL based DSA method that enables collaborative spectrum sharing without requiring users to share their access histories, significantly decreasing the communication overheads and ensuring user privacy. To the best of our knowledge, our work is the first study that utilizes FL in the DRL-based DSA strategies while taking quantized communication into account. 2) Convergence analysis of the Fed-MADRL based DSA algorithm is provided to characterize the trade-off between communication efficiency and system performance. Compared to "zoom in" and "zoom out" quantization in existing FL works, we utilize a more practical quantization method that only allows users to send quantized values to the centralized server. 3) Extensive simulation results show that the Fed-MADRL based DSA method outperforms the independent learning method and achieves comparable performance with the centralized MADRL method, where the performance is evaluated in a realistic DSA scenario of the CBRS system. Furthermore, we utilize an computationally efficient DRL structure to address the continuous state space and partially observable environment in the DSA problem.

The rest of this paper is organized as follows. The system model is described in Section II. The MARL problem formulation is detailed in Section III. The federated training algorithm is explained in Section IV. The convergence analysis is provided in Section V. Simulation results are provided in Section VI. Conclusions are drawn in Section VI. All the important mathematical notations in this paper are listed in Table I

TABLE I: List of notations

Notation	Definition
N	Number of agents (GAA users)
M	Number of channels
$s^n[t]$	State of agent n at time t
$a^n[t]$	Action of agent n at time t
$r^n[t]$	Reward of agent n at time t
$o^n[t]$	Observation of agent n at time t
$s[t]$	Joint state at time t
$a[t]$	Joint action at time t
$r[t]$	Joint reward at time t
γ	Discount factor of RL
P^n	Transmit power of user n
Γ^n	SNR gap of user n
$h_m^{z,n}[t]$	Channel gain between user z's transmitter and user n's receiver on channel m at time t
$q_m[t]$	Activity of channel m at time t
$N_m[t]$	Noise on channel m at time t
$\text{SINR}_m^n[t]$	SINR of user n on channel m at time t
B_m	Bandwidth of channel m
$c_m^n[t]$	Data rate of user n on channel m at time t
$\overline{c}_m^n[t]$	Throughput of user n on channel m until time t
α	Throughput calculation factor m
λ	entropy regularization parameter
T	Time horizon in each local update iteration
K	Number of global update iterations
τ	Number of local update iterations
$\overline{\theta}$	Shared policy network parameters
θ^n	Local policy network parameters of agent n
$f(\cdot)$	Joint value function
$f^n(\cdot)$	Local value function of agent n
η	Local learning rate
β	Global learning rate
$Q(\cdot)$	Quantization function
b	Quantization bits
Δ	Quantization step size
L	Lipschitz constant of value function
Y	Noise of policy gradient
Z	Upper-bound of policy gradient

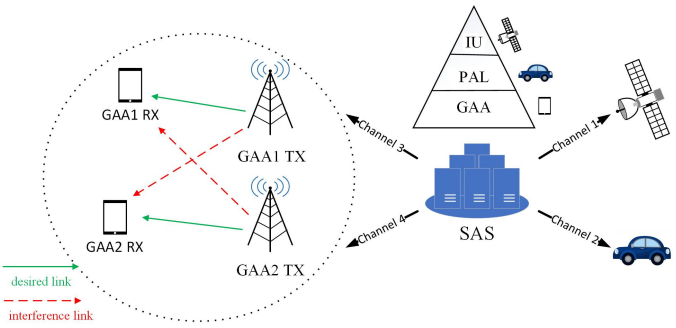

Fig. 1: The CBRS system model.

II. System Model

In this section, we describe the DSA problem in the CBRS system. This article focuses on designing the spectrum access strategies for GAA users to utilize the spectrum resources efficiently. Assume that there are N GAA users sharing M wireless channels, where $1, \cdots, N$ represent the index set of GAA users and $1, \cdots, M$ represent the index set of wireless channels. Most of the literature simplify a GAA user as a node in a graph and design the spectrum access algorithm using graph theory [35]–[37], so the quality-of-service (QoS) of the underlying GAA system cannot be considered. On the other hand, we use the data rate as the QoS of each GAA user and design the spectrum access strategies that maximize the average data rate of all GAA users. Since this paper focuses on studying the performance gain of using FL to solve the DSA problem, we use a relatively simple GAA system model that each GAA user represents a cellular system consisting of a base station and a user equipment (UE) without loss of generality. This work considers downlink transmissions since the UE may not have enough computational power to run the DRL algorithm. Without loss of generality, a GAA user is assumed to be able to access at most one channel at any given time. Accordingly, a one-hot vector $a^n[t] \in \{0,1\}^M$ can be used to denote the index of the accessed channel of GAA user n at time t. Specifically, if GAA user n accesses channel

m at time t, $a^n[t] = \delta_m$, where δ_m is an M-dimensional binary vector whose m^{th} element is equal to 1. It is important to note that GAA users do not receive interference protection from the SAS, so they may interfere with each other if multiple GAA users access the same channel simultaneously. The corresponding data rate of GAA user n on channel m at time t can be characterized as

$$c_m^n[t] = B_m \log_2\left(1 + \frac{\text{SINR}_m^n[t]}{\Gamma^n}\right), \tag{1}$$

where

$$\text{SINR}_m^n[t] = \frac{\mathbb{1}(a^n[t] = \delta_m) P^n \cdot \left|h_m^{n,n}[t]\right|^2}{\sum\limits_{z=1, z\neq n}^{N} \mathbb{1}(a^z[t] = \delta_m) P^z \cdot \left|h_m^{z,n}[t]\right|^2 + N_m[t]}, \tag{2}$$

B_m is the bandwidth of channel m, P^n and P^z are transmit power of GAA user n and GAA user z, respectively, $\mathbb{1}(\cdot)$ is an indicator function, $h_m^{n,n}[t]$ is the channel gain of the desired link between user n's transmitter and receiver at time t, $h_m^{z,n}[t]$ is the channel gain of the interference link between user z's transmitter and user n's receiver at time t, $N_m[t]$ is the noise on channel m at time t, and Γ^n is the SNR gap for the corresponding modulation and coding strategy (MCS) adopted at user n. The SNR gap is an accurate approximation that provides the required SNR for a given modulation and coding strategy to achieve a given target error performance [38]. From the interference term in (2), it can be observed that the data rate of each GAA user on a channel depends on other GAA users that transmit on the same channel. It is important to note that the data rate of GAA users also depend on the activities of higher tier users. According to the spectrum management rules in the CBRS system, a GAA user has to send spectrum access request to the SAS, and the SAS ensures that a GAA user cannot access a channel that is utilized by IUs or PAL users. The channel availability state, $q_m[t] \in \{0, 1\}$, is used to denote the existence of IUs and PAL users on channel m at time t. If $q_m[t]$ equals to 0, then GAA users cannot access channel m because IUs or PAL users are using channel m (channel m is not available). On the other hand, if $q_m[t]$ is equal to 1, channel m is available meaning GAA users are allowed to access channel m. According to spectrum request actions of GAA users and the spectrum coordination by the SAS, we define the reward of GAA user n at time t as the obtained data rate, which is written as

$$r^n[t] = \sum_{m=1}^{M} q_m[t] c_m^n[t]. \tag{3}$$

Note that the received reward is zero if GAA user n sends a spectrum access request to channel m but it is rejected by the SAS.

In many real-world applications, it would be impossible for an agent to perfectly observe the environment and obtain complete state information, so an agent would only receives a partial observation from the environment. We assume that each GAA user utilizes its throughput on channels to decide which channel to transmit in the current time slot. Accordingly, the observation received by GAA user n at time t is defined as

$$o^n[t] = \left(\overline{c}_m^n[t-1]\right), \forall m \in \mathcal{M}, \tag{4}$$

where $\overline{c}_m^n[t-1]$ represents the obtained throughput of user n on channel m until time $t-1$, which is defined as

$$\overline{c}_m^n[t] = \alpha \cdot c_m^n[t-1] + (1-\alpha) \cdot c_m^n[t], \tag{5}$$

where α is between 0 and 1. Since a GAA user's observation is defined as the user throughput, it is affected by the spectrum access behaviors of the higher tier users and other GAA users.

III. MARL Problem formulation

We formulate the DSA problem in the CBRS system as MARL problem where each GAA user is equipped with an RL agent to make its spectrum access decision. MARL is a challenging problem because both the local observation and the local reward received by each agent are influenced by other agents' actions. In other words, an agent not only interacts with the environment but also interacts with other agents, giving rise to a non-stationary environment from each local agent's viewpoint. To be specific, MARL is characterized by N tuples $(\mathcal{S}^n, \mathcal{A}^n, \mathcal{T}^n, R^n, \Omega^n, O^n, \gamma)_{n\in N}$, where N is the number of agents, $\mathcal{S}^n$ is the state space of agent n, $\mathcal{A}^n$ is the action space of agent n, Ω^n is the observation space of agent n, and $\gamma \in [0, 1]$ is the discount factor. At time t, the state of agent n is $s^n[t] \in \mathcal{S}^n$, the observation of agent n is $o^n[t] \in \Omega^n$, the action of agent n is $a^n[t] \in \mathcal{A}^n$, and the reward of agent n is $r^n[t]$. Note that $\mathcal{T}^n$ is the state transition probability of agent n providing $\Pr(s^n[t+1]|s^n[t], a^n[t])$, O^n is the observation probability of agent n providing $\Pr(o^n[t+1]|s^n[t+1], a^n[t])$, R^n is the reward function of agent n providing $r^n[t] = R^n(s^n[t], a^n[t])$. In addition, we denote $\mathcal{S} = \cup_{n=1}^N \mathcal{S}^n$ as the joint state space, $\mathcal{A} = \cup_{n=1}^N \mathcal{A}^n$ as the joint action space, and $R = \frac{1}{N}\sum_{n=1}^N R^n$ as the joint reward function providing $r[t] = R(s[t], a[t]) = \frac{1}{N}\sum_{n=1}^N R^n(s^n[t], a^n[t])$. The goal of MARL is to optimize the cumulative discounted joint reward, which is defined as $\sum_{t=0}^{\infty} \gamma^t r[t]$.

Most MARL algorithms assume a joint reward received by all agents, or individual rewards shared among agents. However, this assumption may not be realistic in many real-world applications due to communication overhead constraints and privacy/security issues. In this paper, we have each agent updates its policy to achieve the goal of maximizing its long-term local reward instead of sharing its local observations or rewards with others. We leverage FL to jointly learn a shared policy that maximizes the joint reward by combining all agents' local policies.

We now formulate the DSA problem using the DRL framework. To be specific, a GAA user n has a DRL agent whose policy network parameters θ^n determines its spectrum access decisions based on its observations. The local state of an agent n at time t can be written as

$$s^n[t] = \left(P^n, \Gamma^n, h_m^{n,n}[t], P^z, h_m^{z,n}[t], a^z[t], N_m[t]\right), \quad \forall m \in \mathcal{M}, \forall z \in \mathcal{N} \setminus n. \tag{6}$$

The joint state at time t can be written as

$$s[t] = \left(P^n, \Gamma^n, h_m^{z,n}[t], a^z[t], N_m[t]\right), \forall n, z \in \mathcal{N}, \forall m \in \mathcal{M}. \tag{7}$$

It can be observed that the joint state space is the union of all agents' local states.

IV. Federated Training Algorithm for DSA

To decrease the communication overheads and increase data privacy, we utilize FL to enable collaborative DSA strategy without requiring GAA users to share their private data.

A. Distributed Federated Policy Gradient

We define agent n's policy π_{θ^n} is obtained from a policy network with parameters θ^n, where π_{θ^n} is a mapping function from agent n's observations to agent n's action decision. Let V^n be the value function of the agent n. Given the agent n's initial state s_i^n and the policy network parameters θ^n, we have

$$V^n\left(s_i^n, \theta^n\right) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R^n(s^n[t], a^n[t]) \mid s^n[0] = s_i^n, a^n[t] \sim \pi_{\pi_{\theta^n}}(s^n[t])\right]. \tag{8}$$

If the agent n's initial state follows a distribution ρ^n, then the agent n's value function becomes a function of θ^n, which is written as

$$f^n(\theta^n) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R^n(s^n[t], a^n[t]) \,\middle|\, s^n[0] \sim \rho^n, a^n[t] \sim \pi_{\theta^n}(s^n[t])\right]. \tag{9}$$

Our goal is to let all agents jointly learn a common policy that can perform well across their environments. To be specific, all agent adopts the same policy to map its local state to its local action, and we utilize FL to enable agents to learn a common policy that maximizes the system goal without sharing each agent's local state, local action, and local reward information. Given that all agents use the same policy π_θ and each agent n's initial state follows a distribution ρ^n, the joint value function can be written as

$$f(\theta) = \frac{1}{N}\sum_{n=1}^{N} f^n(\theta). \tag{10}$$

Accordingly, the goal is to find the parameters θ^* of the joint policy π_{θ^*} that $f(\theta)$ is maximized:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, f(\theta). \tag{11}$$

We utilized the distributed federated policy gradient as the training algorithm, which is formally stated in Algorithm 1. To be specific, each agent first downloads a shared policy network from the centralized controller to set its local policy network. All agents collect their training data by interacting with the environment. To be specific, in the DSA problem, each agent takes the spectrum request access action based on its observations, and then it will receive a reward from the environment. The reward and the observation of each agent are the user data rate and the user throughput, respectively. After collecting T training data in the environment, each agent updates its policy network using the policy gradient method [39]. We let each agent collect data and update its policy network for τ iterations, and then each agent uploads its policy network to the centralized controller. It is important to note that only the policy network is uploaded to the centralized controller, whereas the local training data are kept in each agent. Lastly, the centralized controller aggregates the information of all received local policy networks to update the shared policy network. Then the shared policy network will be downloaded by each local agent again to start a new federated learning cycle. For the DSA problem in the CBRS system, the SAS and GAA users serve as the centralized controller and agents in Algorithm 1.

Algorithm 1 Distributed federated policy gradient.

1: Initialize the local learning rate η for each agent n, the global learning rate β, the quantization function $Q(\cdot)$, and the shared policy network $\overline{\theta}_0$.
2: **for** $k = 1, \cdots, K$ **do**
3: Each agent n downloads the shared policy network to set as its policy network:
$$\theta_{k-1,0}^n = \overline{\theta}_{k-1}, \forall n \in \mathcal{N}.$$
4: **for** $i = 1, \cdots, \tau$ **do**
5: Each user n empties its memory buffer D^n.
6: **for** $t' = 1, \cdots, T$ **do**
7: $t = \tau T(k-1) + T(i-1) + t'$
8: Each agent n receives observation $o^n[t]$ from the environment.
9: Each agent n determines action $a^n[t]$ based on its policy $\pi_{\theta_{k-1,i-1}}$.
10: Each agent receives reward $r^n[t]$ from the environment after executing $a^n[t]$.
11: Each agent n stores $(o^n[t], a^n[t], r^n[t])$ in its memory buffer D^n.
12: **end for**
13: Each agent n computes the policy gradient to update its policy network as follows:
$$\theta_{k-1,i}^n = \theta_{k-1,i-1}^n + \eta \nabla \tilde{f}^n(\theta_{k-1,i-1}^n, D^n). \tag{12}$$
14: **end for**
15: Each agent n uploads a quantized version of its policy network $Q\left(\theta_{k-1,\tau}^n\right)$.
16: The centralized controller updates the shared policy network as follows:
$$\overline{\theta}_k = (1-\beta)\cdot\overline{\theta}_{k-1} + \beta \cdot \frac{1}{N}\sum_{n=1}^{N} Q\left(\theta_{k-1,\tau}^n\right). \tag{13}$$
17: **end for**

To further reduce the communication overheads caused by uploading local policy networks, each agent uploads a quantized version of its policy network using random quantization in [40]. To be specific, we let the weights of policy networks within a finite interval $[l, u]$. The quantization process first

divides this interval into B uniformly spaced bins $[w_0, w_1)$, $\cdots$, $[w_{B-1}, w_B)$, where $w_q - w_{q-1} = \Delta$ for $q = 1, \cdots, B$. Given a weight $x \in [w_{q-1}, w_q)$, the quantization function $Q(x)$ randomly choose w_{q-1} or w_q according to the following distribution:

$$Q(x) = \begin{cases} w_{q-1} & \text{, with probability } 1-p, \\ w_q & \text{, with probability } p, \end{cases} \tag{14}$$

where $p = (x - w_{q-1})/\Delta$. By applying this quantization function to all weights of the policy network, we can obtain a quantized policy network. For a quantization process with B uniformly spaced bins, we need $b = \log_2(B+1)$ bits to represent $w_0, \cdots, w_B$.

B. Applying DRL to DSA

DRL aims to solve the large state space problem in traditional reinforcement learning. Conventional RL techniques such as Q-learning have limited applications with low-dimensional state spaces. DRL utilizes a deep neural network as a function approximator to accelerate the convergence time when the state space is large [41]. In our considered DSA problem, the continuous state results in infinite state space, so we apply DRL as the underlying policy network. However, deploying DRL techniques in DSA systems still has many practical challenges. First, the mobile wireless environments are non-stationary by nature due to many factors, such as user locations, fading, and user traffics. Second, obtaining environment information from the DSA system imposes signalling overhead on network operations. Under these practical issues, the wireless environment of the DSA system is usually non-stationary and partially observable with limited effective training data.

In partially observable environments, an observation at a single time step may not contain sufficient information to predict future rewards and future states. Therefore, a policy that depends on observation histories has better performance in Partially Observable Markov Decision Process (POMDP) environments. A recurrent neural network (RNN) is a powerful neural network structure that can learn the temporal behavior for a time sequence. To deal with the partial observability in many real-world environments, RNNs can be utilized in the DRL to capture the temporal correlation of observation sequences, which is referred to as the deep recurrent Q-network (DRQN) [42]. Even though DRQN is a powerful machine learning tool, it faces serious issues related to training due to two reasons: 1) The kernel of DRQN, the RNN, has issues with vanishing and exploding gradients that make the underlying training computationally inefficient [43]; and 2) DRQN requires a large amount of training data to ensure the learning agent converges to an appropriate policy. Therefore, the difficulties in training DRQN prevent it from being widely adopted in real-world applications [44].

1) Echo State Network: In light of the training challenges of DRQNs, we utilize a special type of RNNs, echo state networks (ESNs), to reduce the training time and the required training data [45]. Given a sequence of inputs $(x[1], \cdots, x[t])$, the update equations of RNN/ESN have the same forms, which are written as:

$$\begin{aligned} h[t] &= \tanh\left(W_{\text{in}} x[t] + W_{\text{rec}} h[t-1]\right), \\ y[t] &= W_{\text{out}} h[t], \end{aligned} \tag{15}$$

where $y[t]$ is the output at time t, $h[t]$ is the hidden state at time t, W_{in} is the input weights, W_{rec} is the recurrent weights, and W_{out} is the output weights. The hidden state $h[t]$ represents a summary of the past sequence of inputs up to t, and we set the initial hidden state $h[0] = \mathbf{0}$. The standard RNN training, backpropagation through time (BPTT), unfolds the network in time into a computational graph that has a repetitive structure to train all weights. On the other hand, ESNs simplify the underlying RNNs training by only training the output weights while leaving input weights and recurrent weights untrained. Specifically, the input weights and the recurrent weights of ESNs are initialized randomly according to the constraints specified by the Echo State Property [46], and then only the output weights of ESNs are trained to accelerate the training speed. The main idea of ESNs is to generate a large reservoir that contains the necessary summary of past input sequences for predicting targets [47]. The learned output weights determine the best linear combination of the reservoir's state and the input signal to perform the desired task. This approach largely reduces the training time because only the output weights are trained. Existing research shows that ESNs can achieve comparable performance with RNNs, especially in some applications requiring fast learning [48].

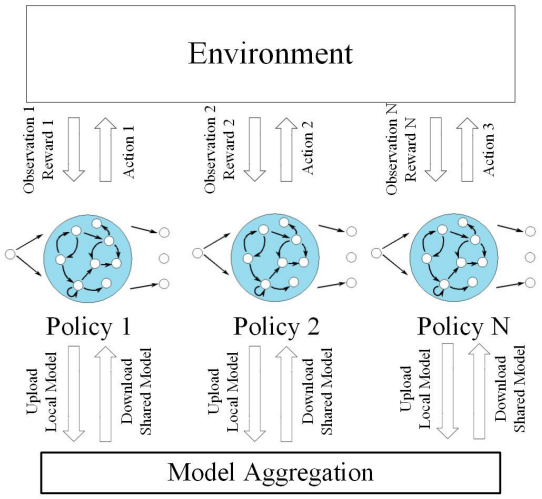

Fig. 2: Fed-MADRL based method using ESN-based policy networks.

2) ESN-based Policy Gradient: To handle the partially observable environment and to accelerate the training, we utilize ESN as the underlying neural network structure of the policy network. Specifically, the softmax function is used as the last activation function of an ESN to generate a probability distribution over the action space. For each agent n, we let the input sequence of ESN be the observation history $o^n_{\leq t} = (o^n[1], \cdots, o^n[t])$, and the parameters of the ESN-based policy network of agent n are $\theta^n = \left(W^n_{\text{in}}, W^n_{\text{rec}}, W^n_{\text{out}}\right)$. The

update equations of the ESN-based policy network is written as:

$$\begin{aligned} y^n[t] &= W^n_{\text{out}} h^n[t], \\ h^n[t] &= \tanh\left(W^n_{\text{in}} o^n[t] + W^n_{\text{rec}} h^n[t-1]\right), \end{aligned} \tag{16}$$

where the dimension of $y^n[t]$ equals to the action space. Accordingly, the resulting distribution over actions at time t is given as

$$\pi_{\theta^n}\left(a|o^n_{\leq t}\right) = \frac{\mathrm{e}^{y^n_a[t]}}{\sum\limits_{a' \in \mathcal{A}} \mathrm{e}^{y^n_{a'}[t]}}, \ \forall a \in \mathcal{A}^n, \tag{17}$$

where $y^n_a[t]$ is the a^{th} element of $y^n[t]$.

From Algorithm 1, the data collected from agent n's environment are $(o^n[t], a^n[t], r^n[t])$ from $t = 1$ to $t = T$. According to the policy gradient algorithm [39], the loss function for training the ESN-based policy is written as

$$\underset{W^n_{\text{out}}}{\operatorname{argmin}} \sum_{t=1}^{T} \left(\log \pi_{\theta^n}\left(a^n[t]|o^n_{\leq t}\right) \cdot \left(\sum_{t'=1}^{t} \gamma^{t'-1} r^n[t'] \right) \right). \tag{18}$$

Due to the exponential scaling of the softmax function, policies may become near deterministic quickly resulting in slow convergence. A common approach is to incorporate entropy regularization into the objective of the policy network [49]. To be specific, we define the entropy-regularized loss function as

$$\begin{aligned} \underset{W^n_{\text{out}}}{\operatorname{argmin}} & \sum_{t=1}^{T} \left(\log \pi_{\theta^n}\left(a^n[t]|o^n_{\leq t}\right) \cdot \left(\sum_{t'=1}^{t} \gamma^{t'-1} r^n[t'] \right) \right) \\ & + \sum_{t=1}^{T} \left(\lambda \cdot \sum_{a \in \mathcal{A}} \pi_{\theta^n}\left(a|o^n_{\leq t}\right) \cdot \log \pi_{\theta^n}\left(a|o^n_{\leq t}\right) \right), \end{aligned} \tag{19}$$

where λ is a regularization parameter.

The system model of the introduced Fed-MADRL based method using ESN-based policy networks is shown in Fig. 2. Each agent first downloads a shared ESN-based policy network from the centralized controller, and then each agent updates the ESN-based policy network locally. Next, each agent only uploads the output weights of the ESN-based policy to the centralized controller because only weights of the ESN-based policy network are trainable. Therefore, the ESN-based policy network is suitable for the FL framework because the communication overheads can be largely decreased.

V. Convergence Analysis

The convergence analysis is conducted under the following assumptions:

Assumption 1. *Each agent n can estimate the unbiased policy gradient $\nabla f^n(\theta)+Y$, where Y is the random noise with $\mathbb{E}[Y] = 0$ and $Var[Y] = \sigma^2$.*

Assumption 2. *Each agent n's value function $f^n(\theta)$ is Lipschitz smooth with constant L:*

$$\left\|\nabla f^n(\theta_1) - \nabla f^n(\theta_2)\right\| \leq L \left\|\theta_1 - \theta_2\right\|.$$

Assumption 3. *Each agent n's policy gradient $\nabla f^n(\theta)$ is upper-bounded: $\|\nabla f^n(\theta)\| \leq Z$.*

Under Assumption 1, $\nabla \tilde{f}^n\left(\theta^n_{k,i-1}, D^n\right) = \nabla f^n\left(\theta^n_{k,i-1}\right) + Y^n_{k,i-1}$, so we can represent the local policy network's update in (12) as

$$\theta^n_{k,i} = \theta^n_{k,i-1} + \eta\left(\nabla f^n\left(\theta^n_{k,i-1}\right) + Y^n_{k,i-1}\right), \quad \forall i \in \{1, \cdots, \tau\}, \tag{20}$$

where

$$\theta^n_{k,0} = \overline{\theta}_k, \tag{21}$$

i is the iteration of local policy update, k is the iteration of FL update, $\theta^n_{k,i}$ is agent n's local policy network at local iteration i and FL iteration k, $\overline{\theta}_k$ is the shared policy network at FL iteration k, η is the local learning rate, β is the global learning rate, and τ is the number of local policy updates. After every τ local policy update, the local policy networks are combined as a single policy network as follows:

$$\overline{\theta}_k = (1-\beta)\cdot\overline{\theta}_{k-1} + \beta \cdot \frac{1}{N}\sum_{n'=1}^{N} Q\left(\theta^{n'}_{k-1,\tau}\right), \quad \forall k \in \{1, \cdots, K\}, \tag{22}$$

where K is the number of FL updates and $\overline{\theta}_0$ is randomly initialized. If $\tau = 1$, then the local policy networks are combined after every local policy update, so the local policy networks are the same as the shared policy network, which is written as

$$\overline{\theta}_k = (1-\beta)\overline{\theta}_{k-1} + \frac{\beta}{N}\sum_{n'=1}^{N} Q\left(\overline{\theta}_{k-1} + \eta\left(\nabla f^{n'}\left(\overline{\theta}_{k-1}\right) + Y^n_{k-1}\right)\right). \tag{23}$$

From (14), the quantized value of $\theta^n_{k,\tau}$ is represented as $Q\left(\theta^n_{k,\tau}\right)$ and the quantization noise is

$$\epsilon^n_{k,\tau} = \theta^n_{k,\tau} - Q\left(\theta^n_{k,\tau}\right), \tag{24}$$

where $\left\|\epsilon^n_{k,\tau}\right\| \leq \Delta$ and $\mathbb{E}\left[\epsilon^n_{k,\tau}\right] = 0$. It is important to note that this quantization noise will cause oscillation in the global update at the centralized server. Therefore, we introduce another step size to mitigate the oscillation in the global update to achieve the convergence of FL under quantization.

Theorem 1. *Suppose that Assumptions 1, 2, and 3 hold. Let the sequence $\{\overline{\theta}_k\}$, for all $k \in [0, K-1]$ be generated by Algorithm 1 and let $f^* = \max_{\theta} f(\theta)$. Then we have*

$$\begin{aligned} \min_{k\in[0,K-1]} \mathbb{E}\left[\left\|\nabla f\left(\overline{\theta}_k\right)\right\|^2\right] \leq & \frac{f^* - \mathbb{E}\left[f\left(\overline{\theta}_0\right)\right]}{\beta\eta\tau K} + L\eta\tau Z(Z+\sigma) \\ & + \frac{L\beta\left(\eta\tau(Z+\sigma)+\Delta\right)^2}{2\eta\tau}. \end{aligned} \tag{25}$$

where the expectation is taken with respect to the quantization.

Proof. See Appendix A. □

Remark 1. *By letting $\beta = O(1/\sqrt{K})$ and $\eta = O(1/K^{1/4})$ and plugging them into (25), we can guarantee that the shared policy network converges to the stationary point at a rate of $O(1/K^{\frac{1}{4}})$. Furthermore, we can characterize the quantization effect on the convergence upper-bound. If the number of quantization bits is smaller, then Δ is larger and*

the upper-bound of (25) is larger. Δ equals to 0 if local policy networks can be uploaded to the centralized server without quantization. This is the trade-off between communication efficiency and system performance. By using fewer bits in the quantization, the communication efficiency improves; however, the system performance may become worse because the output shared policy network may be far from the stationary point. Furthermore, it is important to note that the convergence of FL under quantization (i.e., $\Delta > 0$) cannot be achieved by simply taking the average of local policies (i.e., $\beta = 1$) due to the last term in the upper-bound of (25). On the other hand, when there is no quantized communication (i.e., $\Delta = 0$), the convergence of FL can be achieved at a rate of $O(1/\sqrt{K})$ when $\eta = O(1/\sqrt{K})$ and $\beta = 1$.

VI. Performance Evaluation

A. Experimental Setup

In this section, we evaluate the introduced Fed-MADRL method for the DSA problem in the CBRS system through simulations. There are $N = 8$ GAA users and $M = 4$ channels in a 500m×500m simulation area, where each channel is $B_m = 10$ MHz. The noise spectral density is set to -164 dBm/Hz. We set $\alpha = 0.01$ to calculate the average throughput of each user. Each GAA user can request data transmission on one of the channels, but the final channel allocation is determined by the SAS. In other words, if a GAA user requests a wireless channel that is currently occupied by IUs or PAL users, then it cannot access that wireless channel. For each GAA user n, the distance of the transmitter and the receiver is randomly chosen between 50m to 100m, and the transmit power is set to 50mW. To generate channel gains of desired links and interference links, we set the path loss model as $41 + 22.7 \log_{10}(d)$ dB, where d is the distance between a transmitter and a receiver in meter. The small-scale channel gain follows a Rician distribution, where the ratio of the average power in the line-of-sight path to that in the non-line-of-sight paths is set as 0.8. The dynamics of each channel availability state are modeled as a two-state Markov chain, which is the most widely used model for primary radio user activities. The transition probability of the two-state Markov chain on channel m can be denoted as $\left(p_m^{00}, p_m^{01}, p_m^{10}, p_m^{11}\right)$, where p_m^{ij} represents the probability of the availability of channel m in the next time slot is j given that the current availability state of channel m is i, and thus $p_m^{00} + p_m^{01} = 1$ and $p_m^{10} + p_m^{11} = 1$. For each channel m, we randomly choose p_m^{00} and p_m^{11} from a uniform distribution over $[0.8, 1]$ and $[0, 0.2]$, respectively. Note that our method does not have to know the transition probabilities in priori because we use the model-free RL method.

B. Training Details

We adopt online learning to train the underlying policy networks. To be specific, each GAA user takes channel access actions based on the randomly initialized ESN-based policy network and then utilizes the collected training data to train itself on the fly. All ESN-based policy networks use ESNs with 32 neurons. In each local update iteration, each GAA user updates the output weights of its ESN-based policy network when collecting the sample for $T = 50$ time slots. After τ local update iterations, each GAA user uploads the output weights of its ESN-based policy network to the centralized controller in the SAS. The centralized controller calculates a shared ESN-based policy network by aggregating the information from all local ESN-based policy networks. Accordingly, τ represents the communication delay between the centralized controller and the local agents. Since we stop the simulation when each agent collects 50000 training samples, the number of global update iterations is $K = 50000/(T \cdot \tau) = 1000/\tau$. The learning rates of the local update and the global update are set to $\eta = 0.8$ and $\beta = 0.7$, respectively. For the model quantization, we use $b = 8$ to represent each quantized weight of the ESN-based policy network. Meanwhile, we let the discount factor $\gamma = 0.9$ and the entropy regularization parameter $\lambda = 0.01$.

C. Results

We compare with two baselines: the centralized MADRL method and the independent learning method. On the other hand, we use $\tau = 1$ and $\tau = 5$ for our Fed-MADRL method. In the centralized MADRL method, each agent first sends its training data and policy network to the centralized controller, and the centralized controller trains all policy networks using the training data from all agents. Then the centralized controller sends the updated policy networks back to agents. Since this centralized MADRL method utilizes the information from all agents during training, it can provide a reasonable performance upper-bound of MADRL-based methods. The independent learning method represents that each agent aims to optimize its local reward without sharing information with other agents. We run all our experiments with 100 random seeds, which varied the user geometry and the neural network initialization. The reported curves represent an average over these 100 random seeds, and the shaded areas show the 95% confidence interval.

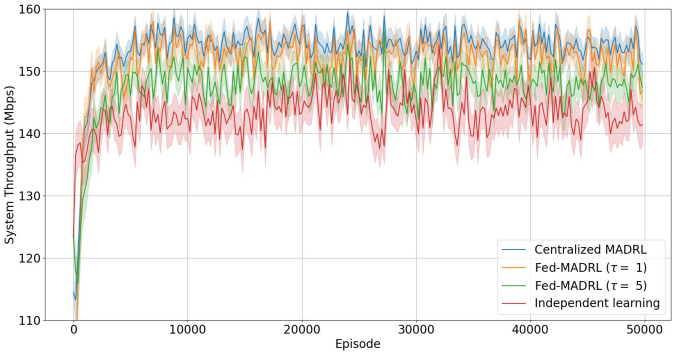

Fig. 3: The system throughput versus time for the centralized MADRL method, the Fed-MADRL method ($\tau = 1$, $\tau = 5$), and the independent learning method.

The curves of the system throughput without quantization are shown in Fig. 3. As expected, the centralized MADRL method achieves the best system performance because it utilizes the information from all users. The Fed-MADRL method with $\tau = 1$ has the second-best performance since

it performs information aggregation after each local policy update. However, the frequent model exchange between the centralized controller and users cause unbearable communication overheads for the DSA network. On the other hand, Fed-MADRL with $\tau = 5$ achieves comparable performance while maintaining reasonable communication overheads between the centralized controller and the local agents. Lastly, the independent learning method has the worst system performance due to no collaboration among agents.

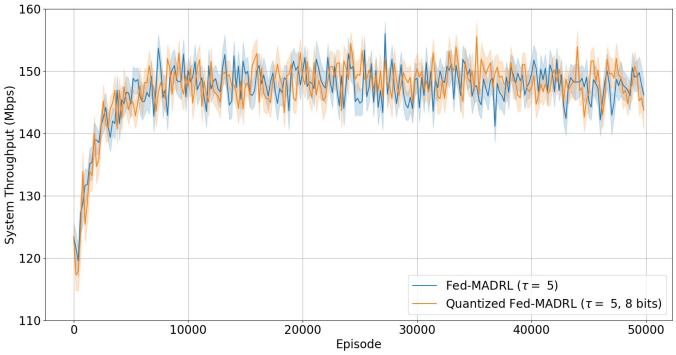

Fig. 4: The effect of quantization.

To further decrease the communication overheads, we only allow the quantized values of policy networks' weights can be exchanged between the centralized controller and GAA users. The effects of random quantization on the Fed-MADRL method are shown in Fig. 4. We can observe that the performance after random quantization is almost no different. Since a float number is usually represented by 16 bits or 32 bits and we only use 8 bits to represent a single value, the communication overheads can be decreased 2 times or 3 times.

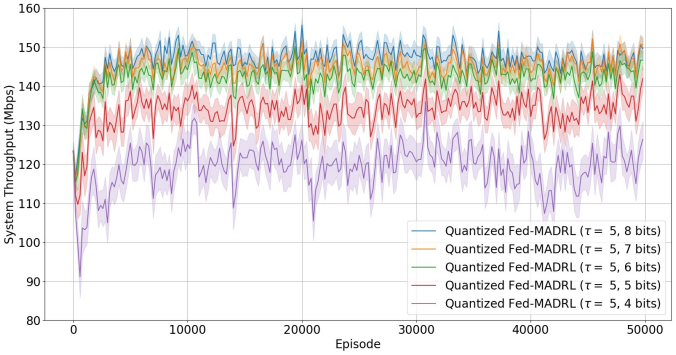

Fig. 5: The effect of the number of quantization bits.

Fig. 5 shows the curves of the system throughput under the different number of quantization bits. Although using fewer quantization bits is more communication-efficient, we can observe that the system performance degrades. The reason is that using fewer bits for quantization reduces the accuracy of uploaded policy networks, and thus it becomes more difficult to achieve the stationary point as shown in Theorem 1. Therefore, there is a trade-off between system performance and communication efficiency.

VII. Conclusion

In this work, we introduce a novel collaborative DSA strategy. To reduce communication overhead and improve data privacy, we utilize FL to design a distributed DSA strategy called Fed-MADRL. FL enables users to learn a joint policy that maximizes the system goal without requiring users to share their private data. We conduct the theoretical analysis to show the trade-off between communication efficiency and system performance. Experimental results show that the introduced Fed-MADRL can achieve comparable performance with the centralized MADRL method which requires global information showing the great promise of applying Fed-MADRL in Beyond 5G and 6G networks.

Appendix A
Proof of Theorem 1

From (20) and (21), it is easily seen that

$$\theta^n_{k,i} = \overline{\theta}_k + \eta \sum_{j=1}^{i} \left(\nabla f^n \left(\theta^n_{k,j-1} \right) + Y^n_{k,j-1} \right), \quad \forall i \in \{1, \cdots, \tau\}. \tag{26}$$

By plugging (24) into (22), we have

$$\overline{\theta}_k = (1-\beta) \cdot \overline{\theta}_{k-1} + \frac{\beta}{N} \sum_{n=1}^{N} \theta^n_{k-1,\tau} - \frac{\beta}{N} \sum_{n=1}^{N} \epsilon^n_{k-1,\tau}. \tag{27}$$

Then we plug (26) into (27) to obtain

$$\begin{aligned} \overline{\theta}_k &= (1-\beta)\overline{\theta}_{k-1} + \beta\overline{\theta}_{k-1} - \frac{\beta}{N} \sum_{n=1}^{N} \epsilon^n_{k-1,\tau} \\ &+ \frac{\beta\eta}{N} \sum_{n=1}^{N} \sum_{j=1}^{\tau} \left(\nabla f^n \left(\theta^n_{k-1,j-1} \right) + N^n_{k-1,j-1} \right) = \overline{\theta}_{k-1} \\ &- \frac{\beta}{N} \sum_{n=1}^{N} \epsilon^n_{k-1,\tau} + \frac{\beta\eta}{N} \sum_{n=1}^{N} \sum_{j=1}^{\tau} \left(\nabla f^n (\theta^n_{k-1,j-1}) + Y^n_{k-1,j-1} \right). \end{aligned} \tag{28}$$

Since $\nabla f^n(\theta)$, $\epsilon^n_{k,\tau}$, and $\mathbb{E}\left[Y^n_{k-1,j-1} \right]$ are upper-bounded, we can obtain

$$\mathbb{E}\left[\left\| \overline{\theta}_k - \overline{\theta}_{k-1} \right\| \right] \leq \beta\eta\tau(Z+\sigma) + \beta\Delta \tag{29}$$

and

$$\begin{aligned} \mathbb{E}\left[\left\| \theta^n_{k,i} - \overline{\theta}_k \right\| \right] &= \mathbb{E}\left[\left\| \eta \left(\sum_{j=1}^{i} \nabla f^n \left(\theta^n_{k,j-1} \right) + N^n_{k,j-1} \right) \right\| \right] \\ &\leq \eta i (Z+\sigma) \leq \eta\tau(Z+\sigma). \end{aligned} \tag{30}$$

We can rewrite (28) as

$$\begin{aligned} \overline{\theta}_k &= \overline{\theta}_{k-1} + \beta\eta\tau\nabla f \left(\overline{\theta}_{k-1} \right) - \frac{\beta}{N} \sum_{n=1}^{N} \epsilon^n_{k-1,\tau} \\ &+ \frac{\beta}{N} \sum_{n=1}^{N} \left(\sum_{j=1}^{\tau} \eta \left(\nabla f^n (\theta^n_{k-1,j-1}) + Y^n_{k-1,j-1} - \nabla f^n (\overline{\theta}_{k-1}) \right) \right) \end{aligned} \tag{31}$$

According to the Lipschitz smooth assumption, we have:

$$f^n\left(\overline{\theta}_k\right) \geq f^n\left(\overline{\theta}_{k-1}\right) + \left\langle \nabla f^n\left(\overline{\theta}_{k-1}\right), \overline{\theta}_k - \overline{\theta}_{k-1} \right\rangle - \frac{L}{2}\left\| \overline{\theta}_k - \overline{\theta}_{k-1} \right\|^2. \tag{32}$$

By plugging (31) into (32), we can obtain

$$f^n\left(\overline{\theta}_k\right) \geq f^n\left(\overline{\theta}_{k-1}\right) + \left\langle \nabla f^n\left(\overline{\theta}_{k-1}\right), \beta\eta\tau \nabla f\left(\overline{\theta}_{k-1}\right) \right\rangle + A^n + B^n + C^n + D, \tag{33}$$

where

$$A^n = \left\langle \nabla f^n(\overline{\theta}_{k-1}), \frac{\beta\eta}{N}\sum_{n=1}^{N}\sum_{j=1}^{\tau}\left(\nabla f^n(\theta^n_{k-1,j-1}) - \nabla f^n(\overline{\theta}_{k-1})\right) \right\rangle \tag{34}$$

$$B^n = \left\langle \nabla f^n\left(\overline{\theta}_{k-1}\right), -\frac{\beta}{N}\sum_{n=1}^{N}\epsilon^n_{k-1,\tau} \right\rangle \tag{35}$$

$$C^n = \left\langle \nabla f^n\left(\overline{\theta}_{k-1}\right), \frac{\beta\eta}{N}\sum_{n=1}^{N}\sum_{j=1}^{\tau} Y^n_{k-1,j-1} \right\rangle \tag{36}$$

$$D = -\frac{L}{2}\left\| \overline{\theta}_k - \overline{\theta}_{k-1} \right\|^2. \tag{37}$$

Since $\mathbb{E}\left[\epsilon^n_{k-1,\tau}\right]$ and $\mathbb{E}\left[Y^n_{k-1,j-1}\right]$ equal to zeros, B^n and C^n can be eliminated by taking expectation on both sides of (33) as follows:

$$\mathbb{E}\left[f^n\left(\overline{\theta}_k\right)\right] \geq \mathbb{E}\left[f^n\left(\overline{\theta}_{k-1}\right)\right] + \mathbb{E}\left[A^n\right] + \mathbb{E}\left[D\right] + \beta\eta\tau\mathbb{E}\left[\left\langle \nabla f^n\left(\overline{\theta}_{k-1}\right), \nabla f\left(\overline{\theta}_{k-1}\right)\right\rangle\right]. \tag{38}$$

Using the Lipschitz smooth assumption and (30), we have

$$\mathbb{E}\left[\left\|\nabla f^n(\theta^n_{k-1,j-1}) - \nabla f^n(\overline{\theta}_{k-1})\right\|\right] \leq L\mathbb{E}\left[\left\|\theta^n_{k-1,j-1} - \overline{\theta}_{k-1}\right\|\right] \leq L\eta\tau(Z+\sigma). \tag{39}$$

We use the Cauchy–Schwarz inequality to upper bound $|A^n|$ as follows:

$$\mathbb{E}\left[|A^n|\right] \leq \left\|\nabla f^n(\overline{\theta}_{k-1})\right\| \mathbb{E}\left[\left\|\frac{\beta\eta}{N}\sum_{n=1}^{N}\sum_{j=1}^{\tau}\left(\nabla f^n(\theta^n_{k-1,j-1}) - \nabla f^n(\overline{\theta}_{k-1})\right)\right\|\right] \leq Z\cdot\frac{\beta\eta}{N}\cdot N\cdot\tau\cdot L\eta\tau(Z+\sigma) = L\eta^2\tau^2 Z(Z+\sigma)\beta. \tag{40}$$

Thus we have

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[A^n\right] \geq -L\eta^2\tau^2 Z(Z+\sigma)\beta. \tag{41}$$

From (29), we can upper bound $\mathbb{E}\left[D\right]$ as

$$\mathbb{E}\left[D\right] \geq -\frac{L\left(\beta\eta\tau(Z+\sigma)+\beta\Delta\right)^2}{2}. \tag{42}$$

Summing up both sides of (38) over n, we can obtain

$$\mathbb{E}\left[f\left(\overline{\theta}_k\right)\right] \geq \mathbb{E}\left[f\left(\overline{\theta}_{k-1}\right)\right] + \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[A^n\right] + \mathbb{E}\left[D\right] + \frac{\beta\eta\tau}{N}\sum_{n=1}^{N}\mathbb{E}\left[\left\langle \nabla f^n\left(\overline{\theta}_{k-1}\right), \nabla f\left(\overline{\theta}_{k-1}\right)\right\rangle\right] = \mathbb{E}\left[f\left(\overline{\theta}_{k-1}\right)\right] + \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[A^n\right] + \mathbb{E}\left[D\right] + \beta\eta\tau\mathbb{E}\left[\left\langle \nabla f\left(\overline{\theta}_{k-1}\right), \nabla f\left(\overline{\theta}_{k-1}\right)\right\rangle\right]. \tag{43}$$

By using the upper-bounds in (41) and (42), we have

$$\beta\eta\tau\mathbb{E}\left[\left\|\nabla f\left(\overline{\theta}_{k-1}\right)\right\|^2\right] \leq \mathbb{E}\left[f\left(\overline{\theta}_k\right)\right] - \mathbb{E}\left[f\left(\overline{\theta}_{k-1}\right)\right] + L\eta^2\tau^2 Z(Z+\sigma)\beta + \frac{L\left(\beta\eta\tau(Z+\sigma)+\beta\Delta\right)^2}{2}. \tag{44}$$

From (44), we can obtain

$$\frac{\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f\left(\overline{\theta}_k\right)\right\|^2\right]}{K} \leq \frac{\mathbb{E}\left[f\left(\overline{\theta}_K\right)\right] - \mathbb{E}\left[f\left(\overline{\theta}_0\right)\right]}{\beta\eta\tau K} + L\eta\tau Z(Z+\sigma) + \frac{L\beta\left(\eta\tau(Z+\sigma)+\Delta\right)^2}{2\eta\tau} \leq \frac{f^* - \mathbb{E}\left[f\left(\overline{\theta}_0\right)\right]}{\beta\eta\tau K} + L\eta\tau Z(Z+\sigma) + \frac{L\beta\left(\eta\tau(Z+\sigma)+\Delta\right)^2}{2\eta\tau}. \tag{45}$$

Then we have

$$\min_{k\in[0,K-1]}\mathbb{E}\left[\left\|\nabla f\left(\overline{\theta}_k\right)\right\|^2\right] \leq \frac{f^* - \mathbb{E}\left[f\left(\overline{\theta}_0\right)\right]}{\beta\eta\tau K} + L\eta\tau Z(Z+\sigma) + \frac{L\beta\left(\eta\tau(Z+\sigma)+\Delta\right)^2}{2\eta\tau}. \tag{46}$$

REFERENCES

[1] Cisco public, "Cisco annual internet report (2018–2023)," Cisco Systems Inc., San Jose, CA, USA, Tech. Rep., Mar. 2020.

[2] R. Bacchus, T. Taher, K. Zdunek, and D. Roberson, "Spectrum utilization study in support of dynamic spectrum access for public safety," in *IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*, 2010, pp. 1–11.

[3] V. Valenta, R. Maršálek, G. Baudoin, M. Villegas, M. Suarez, and F. Robert, "Survey on spectrum utilization in Europe: Measurements, analyses and observations," in *IEEE Intl. Conf. Cogn. Radio Oriented Wireless Networks Commun. (CROWNCOM)*, 2010, pp. 1–5.

[4] S. Bhattarai, J.-M. J. Park, B. Gao, K. Bian, and W. Lehr, "An overview of dynamic spectrum sharing: Ongoing initiatives, challenges, and a roadmap for future research," *IEEE Trans. Cogn. Commun. Networking*, vol. 2, no. 2, pp. 110–128, 2016.

[5] J. Lee and H.-K. Park, "Channel prediction-based channel allocation scheme for multichannel cognitive radio networks," *KICS J. Commun. Networks*, vol. 16, no. 2, pp. 209–216, 2014.

[6] V. K. Tumuluru, P. Wang, and D. Niyato, "Channel status prediction for cognitive radio networks," *Wireless Commun. Mobile Comput.*, vol. 12, no. 10, pp. 862–874, 2012.

[7] X. Li and S. A. Zekavat, "Cognitive radio based spectrum sharing: Evaluating channel availability via traffic pattern prediction," *KICS J. Commun. Networks*, vol. 11, no. 2, pp. 104–114, 2009.

[8] Y. Wang, Z. Ye, P. Wan, and J. Zhao, "A survey of dynamic spectrum allocation based on reinforcement learning algorithms in cognitive radio networks," *Artificial intelligence review*, vol. 51, no. 3, pp. 493–506, 2019.

[9] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, 2018.

[10] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Trans. Cogn. Commun. Networking*, vol. 4, no. 2, pp. 257–265, 2018.

[11] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Select. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, 2019.

[12] A. Doshi, S. Yerramalli, L. Ferrari, T. Yoo, and J. G. Andrews, "A deep reinforcement learning framework for contention-based spectrum sharing," *IEEE J. Select. Areas Commun.*, 2021.

[13] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, 2019.

[14] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, 2019.

[15] X. Huang, S. Leng, S. Maharjan, and Y. Zhang, "Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9282–9293, 2021.

[16] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in *IEEE Intl. Conf. Computer Commun.*, 2013, pp. 2751–2759.

[17] H. Li, Y. Yang, Y. Dou, J.-M. J. Park, and K. Ren, "PeDSS: Privacy enhanced and database-driven dynamic spectrum sharing," in *IEEE Intl. Conf. Computer Commun.*, 2019, pp. 1477–1485.

[18] H.-H. Chang, H. Song, Y. Yi, J. Zhang, H. He, and L. Liu, "Distributive dynamic spectrum access through deep reinforcement learning: A reservoir computing-based approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1938–1948, 2018.

[19] C. Zhong, Z. Lu, M. C. Gursoy, and S. Velipasalar, "A deep actor-critic reinforcement learning framework for dynamic multichannel access," *IEEE Trans. Cogn. Commun. Networking*, vol. 5, no. 4, pp. 1125–1139, 2019.

[20] H.-H. Chang, L. Liu, and Y. Yi, "Deep echo state Q-network (DEQN) and its application in dynamic spectrum sharing for 5G and beyond," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020.

[21] L. Li, L. Liu, J. Bai, H.-H. Chang, H. Chen, J. D. Ashdown, J. Zhang, and Y. Yi, "Accelerating model-free reinforcement learning with imperfect model knowledge in dynamic spectrum access," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7517–7528, 2020.

[22] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, 2017, pp. 1273–1282.

[23] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, 2020.

[24] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, 2020.

[25] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, 2019.

[26] P. Tehrani, F. Restuccia, and M. Levorato, "Federated deep reinforcement learning for the distributed control of NextG wireless networks," in *IEEE Intl. Symposium on Dynamic Spectrum Access Networks*, 2021, pp. 248–253.

[27] M. Krouka, A. Elgabli, C. B. Issaid, and M. Bennis, "Communication-efficient and federated multi-agent reinforcement learning," *IEEE Trans. Cogn. Commun. Networking*, 2021.

[28] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *Intl. Conf. Machine Learn.*, 2019, pp. 2961–2970.

[29] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Intl. Conf. Machine Learn.*, 2018, pp. 5872–5881.

[30] H. Ryu, H. Shin, and J. Park, "Multi-agent actor-critic with hierarchical graph attention network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 7236–7243.

[31] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *Intl. Conf. Artif. Intell. Stat.*, 2020, pp. 2021–2031.

[32] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," *Adv. Neural Inf. Processing Syst.*, vol. 30, 2017.

[33] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, "Quantized federated learning under transmission delay and outage constraints," *IEEE J. Select. Areas Commun.*, vol. 40, no. 1, pp. 323–341, 2021.

[34] Federal Communications Commission, "Amendment of the commission's rules with regard to commercial operations in the 3550-3650 mhz band," *Report and Order and Second Further Notice of Proposed Rulemaking, GN Docket*, no. 12-354, 2015.

[35] X. Ying, M. M. Buddhikot, and S. Roy, "SAS-assisted coexistence-aware dynamic channel assignment in cbrs band," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6307–6320, 2018.

[36] Ł. Kułacz, P. Kryszkiewicz, A. Kliks, H. Bogucka, J. Ojaniemi, J. Paavola, J. Kalliovaara, and H. Kokkinen, "Coordinated spectrum allocation and coexistence management in cbrs-sas wireless networks," *IEEE Access*, vol. 7, pp. 139 294–139 316, 2019.

[37] J. Jeon, R. D. Ford, V. V. Ratnam, J. Cho, and J. Zhang, "Coordinated dynamic spectrum sharing for 5G and beyond cellular networks," *IEEE Access*, vol. 7, pp. 111 592–111 604, 2019.

[38] A. Garcia-Armada, "SNR gap approximation for M-PSK-based bit loading," *IEEE Trans. Wireless Commun.*, vol. 5, no. 1, pp. 57–60, 2006.

[39] S. M. Kakade, "A natural policy gradient," *Adv. Neural Inf. Processing Syst.*, vol. 14, 2001.

[40] T. C. Aysal, M. J. Coates, and M. G. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Trans. Signal Processing*, vol. 56, no. 10, pp. 4905–4918, 2008.

[41] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[42] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," in *AAAI Fall Symposium Series*, 2015.

[43] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Intl. Conf. Machine Learn.*, 2013, pp. 1310–1318.

[44] G. Dulac-Arnold, D. Mankowitz, and T. Hester, "Challenges of real-world reinforcement learning," *arXiv preprint arXiv:1904.12901*, 2019.

[45] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," German National Research Center For Information Technology, Tech. Rep. 34, Jan. 2001.

[46] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 659–686.

[47] P. Enel, E. Procyk, R. Quilodran, and P. F. Dominey, "Reservoir computing properties of neural dynamics in prefrontal cortex," *PLoS computational biology*, vol. 12, no. 6, p. e1004967, 2016.

[48] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: a review," *Neural Networks*, 2019.

[49] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "On the theory of policy gradient methods: Optimality, approximation, and distribution shift," *J. Machine Learn. Res.*, vol. 22, no. 98, pp. 1–76, 2021.

Hao-Hsuan Chang received the Ph.D. degree in electrical and computer engineering from Virginia Tech, USA, in 2021, the M.S. degree in communication engineering from National Taiwan University, Taiwan, in 2015, and the B.Sc. degree in electrical engineering from National Taiwan University, Taiwan, in 2013. He is currently the Senior Research Engineer in Samsung Research America. His research interests include dynamic spectrum access, deep reinforcement learning, and machine learning for wireless communications.

Yifei Song received his B.S. degree in Electrical Engineering and M.S. in Electrical and Computer Engineering from University of Connecticut and University of Washington respectively in 2017 and 2020. He joined the department of Electrical and Computer Engineering at Virginia Tech as a Ph. D. student in 2021 Spring. His current research interests are in the broad area of wireless communications, machine learning and optimization.

Thinh T. Doan is an Assistant Professor in the Department of Electrical and Computer Engineering at Virginia Tech. He obtained his Ph.D. degree at the University of Illinois, Urbana-Champaign, his master degree at the University of Oklahoma, and his bachelor degree at Hanoi University of Science and Technology, Vietnam, all in Electrical Engineering. His research interests span on the intersection of control theory, optimization, machine learning, game theory, and applied probability.

Lingjia Liu received his Ph.D. degree in electrical and computer engineering from Texas A&M University, USA and his B.S. in electronic engineering from Shanghai Jiao Tong University. Currently, he is a professor in the Bradley Department of Electrical and Computer Engineering at Virginia Tech, USA. He is also serving as the Director of Wireless@Virginia Tech. His research interests include machine learning for wireless communications, enabling technologies for 5G and beyond, mobile edge computing, and Internet of Things.