Scalable Multi-Agent Reinforcement Learning for Dynamic Coordinated Multipoint Clustering

Fenghe Hu, Yansha Deng, *Senior Member, IEEE*, and A. Hamid Aghvami, *Life Fellow, IEEE*

***Abstract*— Reinforcement learning (RL) is a widely investigated intelligent algorithm and proved to be useful in the wireless communication area. However, for optimization problems in large-scale multi-cell networks whose dimension increases exponentially, it is unrealistic to employ a conventional centralized RL algorithm and make decisions for the entire network. Multi-agent RL, which allows distribute decision-making, is expected to solve the scalability problem but with performance issues due to the unknown global information, i.e., non-stationary environment. In this paper, we propose a parameter-sharing multi-agent RL for grouping decisions of coordinated multi-point in a large-scale network, where agents jointly serve users to enhance the cell-edge service. By sharing information via parameters, our theoretical and simulation results show that parameter sharing can largely benefit the multi-agent algorithm with convergence proof and convergence speed analysis. To reduce the effect of biased local heterogeneous experience, we also propose a transfer learning method for the parameter sharing process, whose performance of transfer learning algorithms is verified by the simulation results.**

***Index Terms*— Reinforcement learning, telecommunication, multi-agent systems, next generation networking.**

I. Introduction

As A proposed research direction in future wireless networks, the densification of communication network calls for cognitive/self-organized network management for interference, coordination, and operation expenditures. Currently, the performance gain of the densification is mostly achieved by developing interference tolerant methods or shutting down certain nodes when possible, but the coordination between wireless accesspoints (APs) can effectively reduce the interference. Following this idea the concept of a coordinated multipoint (CoMP) or cell-free (CF) network is introduced recently. In CoMP, by coordinate neighbouring APs to serve cell-edge users, the strongest interference term is turned as signal [1]. CF network allows the entire network to serve as distributed antenna system, which removes the concept of cell [2].

Though the APs coordination are a promising technology to enhance the gain of dense deployment. The coordination can cause additional cost of frequency resource and fronthaul/backhaul capacity, and load imbalance [2]. It is necessary to carefully design the clustering algorithm. There are three major clustering schemes: static, semi-dynamic, and dynamic clustering. The static clustering provides a fixed coordination scheme which is simple and basic without the need for communication between APs [3]. However, it fails to continue optimising in a dynamic environment. By allowing adjustments in the static scheme, semi-dynamic clustering can capture part of environmental characteristics and adjust the network with reasonable complexity [4]. It is always desirable to consider full dynamic clustering. However, the calculation of an optimised clustering scheme can cause significant computation overhead, whose complexity increases exponentially with the network scale. That is the so-called high-dimensional or scalability problem, which limits the development of intelligent large-scale networks. Similar scenarios also include: radio resource management (RRM) [5], [6], unmanned aerial vehicle (UAV) route management [7], mobile edge computing (MEC) resource management [8]. Existing solutions for the dynamic clustering problem mainly use greedy and game-theory algorithms [2], but usually lead to sub-optimal solutions. Recently, many intelligent algorithms are developed to optimise network performance, including reinforcement learning (RL). With the capability of learning from environmental experience, RL is a suitable algorithm for wireless communication scenarios [9]. RL algorithms contain agents, policies, and the environment. Agents are trained to perform sequential decisions through policies by interacting with the environment. Commonly, RL approaches are centralized, which employs a centralized agent responsible for all decisions inside the network. Such a centralized structure can cause significant overhead in backbone transmission and computation occupancy. The recent achievement in solving the scalability problem with RL, i.e. Go, can be barely applied in the wireless communication area, as the precise simulation of possible future states required by the decision tree is unrealistic and the computation expanse won't pay off.

To fully realize the benefit of RL for the large-scale network, researchers have tried to distribute the action decisions to entities inside the network, i.e., multi-agent RL (MARL). Similar to greedy or game theory algorithms, each agent (e.g., APs in CoMP scenario) solves its decomposed sub-problem distributively while still optimizing the common global target, where the overall complexity and computation expanse can be reduced. With these advantages, researchers have tried MARL in scenarios with scalability problems, e.g. CoMP [2]. However, unlike single-agent RL, multi-agent RL suffers from convergence and instability problems due to the non-stationary environment, which is caused by unknown information from

Manuscript received 29 March 2022; revised 4 September 2022; accepted 23 October 2022. Date of publication 9 November 2022; date of current version 16 January 2023. This work was supported by Engineering and Physical Sciences Research Council (EPSRC), U.K., under Grant EP/W004348/1. The associate editor coordinating the review of this article and approving it for publication was M. C. Gursoy. *(Corresponding author: Yansha Deng.)*

The authors are with the Engineering, King's College London, London WC2R 2LS, U.K. (e-mail: fenghe.hu@kcl.ac.uk; yansha.deng@kcl.ac.uk; hamid.aghvami@kcl.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2022.3220870.

Digital Object Identifier 10.1109/TCOMM.2022.3220870

other agents. Thus, directly allowing agents to make their decisions based on local observations usually result in limited performance [8]. The more information about neighbours available in agents, the higher performance the algorithm can achieve. Without any information, opponent modelling is an intuitive solution [10]. The agents have to guess the possible action of their opponents, and vice versa. This can create an infinitive logic guessing loop. Another feasible solution is collecting experience from all agents and training a global policy conditioned on local observations. The global policy can be executed distributively among agents, namely, centralized-training-distributive-execution (CTDE) [5]. This approach is widely used by researchers in the communication area and usually achieves reasonable result [11, Table. V]. However, this approach still requires stable connectivity among agents due to the heavy backhaul load of continually experiencing transmission. Another approach is training a critic with full knowledge of the environment conditioned on all agent's decisions, which is then used to guide the training of policy distributively at each agent, i.e. MADDPG [9]. This approach allows the decision to be made by agents locally. It achieves good performance but is still limited by scalability. The decision can be made distributively without communication, but the algorithm requires global reward estimator for each action. This problem has been investigated in recent wireless communication researches [5], [8], [12], but they only allows limited number of agents (mostly below 4) in their simulations. The work [13] applied a value decomposition network to scale the estimation part, which needs the assumption of a fully decomposed value function without overlapping between local values. Similarly, QMIX relaxes this assumption and achieves good performance by combining the output of the value function via a combination network referenced on global information [14], [15]. It is obvious that for multi-agent RL, the more global information, the better performance.

To ensure the performance of multi-agent algorithms, it is necessary to find an effective way to exchange information while solving the scalability problem [11]. The researchers have tried different sharing schemes, including the abstract policy, value function, action advice, and the prediction models [16]. They discussed suitable solution for wireless scenarios and obtained useful results. The wireless network can be abstracted as a graph [6] and information sharing might be only required between neighbouring agents to solve wireless communication problems. Graph abstraction is reasonable with the wireless signal. As each user or AP has a limited coverage area as the signal fades with the increasing of distance, and the delay increases with the travelling distance and hops. This naturally limits the effective region of certain agents by its constrained coverage range and connected neighbours. The overlapped agents' effective regions describe their impact on neighbours and can be seen as networked and connected to their neighbours. The quantification of such a connection's strength and identification of the topology structure is difficult in wireless scenarios. A similar approach quantifies the strength via mean-field theorem [17].

Inspired by the aforementioned ideas, we propose a multi-agent structure with parameter sharing methods for the network-centric clustering in CoMP. Similar but different to CTDE, sharing the network parameters instead of experience (e.g., observations, actions) saves backhaul capacity. We exchange the network parameters to share knowledge among agents while keeping the training locally by sharing parameters. To measure the impact of neighbours, we apply a convolutional neural network (CNN) to abstract features from the environment [18]. We show that the intention can be naturally acknowledged by neighbours following the common features in overlapped effective regions via shared parameters in CNN. In this paper, we introduce a parameter-sharing multi-agent RL to address the grouping problem in the cooperative transmission. Each agent decides their cooperate decision reference on their effective region individually while sharing the parameters of their decision policy globally. The contributions of this paper are summed as follows:

- We propose a parameter sharing distributive multi-agent RL algorithm for a coordinated multi-point (CoMP) problem, which faces a high dimensional grouping problem. Combining the characteristics of wireless communication, we provide convergence proof to show the benefit of parameter sharing in a wireless communication multi-agent system.
- As parameter sharing requires backhaul traffic which is critical for wireless systems, we derive the upper bound of convergence speed for our proposed parameter sharing multi-agent RL with the informational model and show the relationship between parameter sharing frequency and convergence speed.
- We highlight the remaining problem of centralized-decentralized-mismatch [19] in the parameter sharing system and analyze the impact of this phenomenon. We also apply the correlation alignment method which keeps local experience characteristics while sharing parameters.
- We implement our proposed parameter sharing multi-agent RL with different state-of-the-art RL algorithms in a simulated CoMP scenario [20], [21]. We show that our proposed parameter-sharing multi-agent algorithms can effectively solve the grouping problem in CoMP, whose problem dimension exceeds $3e21$. Especially, transfer learning is shown helpful in reducing the impact of centralized-decentralized-mismatch significantly. We also verify our analysis results of the convergence speed versus parameter sharing frequency via simulation.

II. A Coordinated Multipoint Wireless Networks Model

In this section, to ease the understanding of the basics and showcase the benefits of our proposed algorithm, we first present the system model of a joint-transmission coordinated multipoint (JT-CoMP) in a large-scale multi-cell network. It groups APs to enhance the service quality of cell-edge users and currently faces scalability problems in finding grouping policy [22].

A. System Model

We consider a network-supported JT-CoMP for downlink transmission with a set of APs, denoted by $\mathcal{B}$. For simplicity,

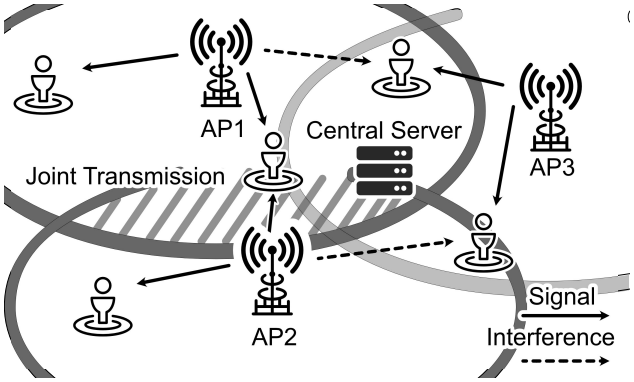

Fig. 1. JT-CoMP Scenario with AP1 and AP2 forming a joint transmission group to enhance the service for users in the overlapped effective area while causing interference to users serving by AP3.

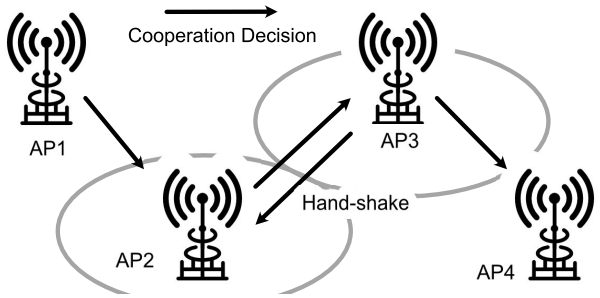

Fig. 2. The handshake process for group forming. The AP2 and AP3 form a coordination group by sending requests to each other, while AP1 and AP4 fail to form a group without the agreement from the AP2 and AP3.

each AP is located in a hexagonal grid and equipped with one omnidirectional antenna. All APs are connected to a central server via fibre backhaul links, which allows the sharing of control signals and data. A set of users, denoted by $\mathcal{U}$, locate in the serving area following the Poisson clustered process (PCP) distribution as it is realistic compared to the Poisson point process (PPP) whose details are discussed later.

Through joint transmission, the CoMP technology enhances the cell-edge users' QoS at the cost of backhaul overhead and frequency resource [23]. As shown in Fig. 1, the neighbouring APs seek to form cooperative groups [22], where the signals are transmitted and enhanced by both cooperative APs using the same frequency band. In particular, the users benefit from the large group and effective collaboration among APs. The larger the group size, the more effective cooperation among APs, and the higher backhaul capacity and frequency resource requirements. The collaboration requires reserving frequency bands for cell-edge users in both APs [23]. Besides, the synchronization among APs is also challenging. Due to these limitations, it is common to have a limited number of cooperative APs, i.e., maximum group size, as $B_{\max}$ [1].

With such a network, we consider that users randomly arrive inside the service area continually following certain distribution every T_u. Upon arrival, each user requests a certain amount of data D from APs in the downlink. The request is considered to be failed if it is not satisfied within a certain period T. The user is removed from the environment after its request is satisfied or failed. Then, we quantify the service performance of the network with a certain cooperation policy π via a QoS function, which is usually a function of the resulting users' signal-to-interference-noise-ratio. We adopt spectrum efficiency with the consideration of service outage as the QoS function for considered CoMP scenario [2]

$$r_t^u = \log_2(1 + \underbrace{\frac{\sum_{b\in\mathcal{B}^u} P_b|w_{b,u}h_{b,u}|^2}{\sum_{b'\in\mathcal{B}^{\mathcal{U}/u}} P_{b'}|w_{b',u'}h_{b',u}|^2 + \sigma^2}}_{\text{SINR}_u}), |\{\mathcal{B}^u\} \sim \pi, \tag{1}$$

where P_b is the transmit signal power from b-th AP, $h_{b,u}$ is the path loss from b-th AP to u-th user, i.e. $h_{b,u} = \beta_{b,u} d_{b,u}^{-\alpha}$, $\beta_{b,u}$ is the small scale factor, $d_{b,u}^{-\alpha}$ present the large-scale fading which is considered as free-space fading for simplification, $w_{b,u}$ is the beamforming vector, $d_{b,u}^{-\alpha}$ denotes the large-scale fading that depends on the distance and path loss factor α. $\mathcal{B}^u$ is the set of cooperative APs associated with user u, which is controlled by both the cooperation policy π and the close-first association scheme. User u is associated with the closest AP, while the other APs in the same cooperative group jointly serve u, i.e. $\mathcal{B}^u$. $\mathcal{B}^{\mathcal{U}/u}$ is the set of other APs not serving user u, σ^2 is the noise power.

To fully distribute coordination decisions among the network, we consider a hand-shake mechanism in our model to allow agents fully decide and control their cooperation. Each AP sends a request to cooperate with certain neighbouring APs. The cooperation group is formed among APs agreeing on the cooperation via hand-shake, as shown in Fig.2. By performing the hand-shake, the cooperation groups for the whole network ($\mathcal{B}^u$) are decided via one single round-trip information exchange between neighbouring agents. The design of the hand-shake enhances the scalability but requires awareness of neighbours' possible actions to ensure successful cooperation.

For the considered cooperative CoMP problem, the policy π decides the cooperation decisions of all APs referenced on the environment and service status. Following the definition of QoS in Eq.(1), the optimization target of the considered CoMP network is the long-term summed QoS

$$\max_{\pi} \sum_t^{\infty} [\sum_{u\in\mathcal{U}} r_t^u | \{\mathcal{B}^u\}_{u\in\mathcal{U}} \sim \pi], s.t. \quad |\mathcal{B}^u| \le B_{\max} \tag{2}$$

where r_t^u is defined in Eq.(1), $B_{\max}$ is the maximum group size, which is usually around 3 [22].

Commonly for our considered optimization problem, grouping decisions, i.e., $\mathcal{B}^u$, are decided by a central utility referenced on information collected from all APs. The optimization of such cooperation problems increases with the number of cooperative APs, which is shown to be NP-hard [24]. Apart from the scalability problem mentioned before, this process consumes at least a round-trip delay from individual APs to the central utility, whose decisions might be outdated upon arrival. In the following section, we decompose the problem geometrically and solve it distributively.

B. Problem Decomposition via Effective Region

Inspired by the idea of solving the original large-scale problem distributively, designing a distributive algorithm with

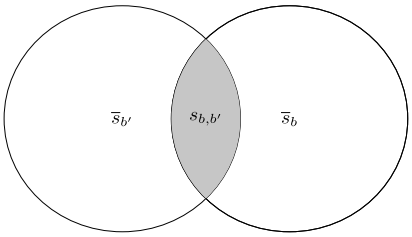

Fig. 3. The relationship between overlapped effective region $s_{b,b'}$ and $\hat{s}_b$ for two neighboring APs.

intelligent methods is a straightforward idea, but usually results in undesired performance. In the following section, we try to identify several key properties of problem decomposition in a large-scale multi-cell wireless network, which allows special designs of intelligent algorithms.

We first highlight that the QoS function in Eq. (1) can be considered geometrically separable and independent. The QoS value only depends on the states of local users (the SINR of user u), which is affected by the cooperation decisions and the signal strength received at that location. In this way, the QoS value in different locations is independent of each other. Second, the wireless signals fade with the increasing distance or the existence of variant obstacles. Each AP imposes limited signal gain/interference to the surrounding area. Thus, each AP has an effective region that is limited by its maximum coverage area. The QoS value of the effective region precisely reflects the performance of local APs' and neighbours' policies. Here, we assume that all APs have similar technical specifications. Instead of pico, micro and macro cells, all APs are overlapped but are not covered effective region. The multi-agent algorithms for heterogeneous cellular networks are too complex and out of scope for this paper. We define the effective region of bth AP as the set of users inside the bth AP's effective region, denoted as $\mathcal{U}^b$:

$$\mathcal{U}^b = \{u | P_b d_{b,u}^{-\alpha} \geq \sigma\} \quad \forall u \in \mathcal{U}, \tag{3}$$

where only users close enough are considered inside the effective region, σ is the average receive power threshold of the considering uth user is in bth AP's effective region [23], and we have $\mathcal{U}^b \subset \mathcal{U}$. For simplicity, the effective region can be considered as a circle because of line-of-sight transmission. The irregular effective region of the unreachable or poor effective region can be naturally described by the users' position following Eq.(3). But this can cause heterogeneous experiences in different APs which will be discussed later.

Within each AP's effective region, we can further decompose the local QoS function into one region, which is majorly affected by the AP itself, and other regions that are jointly affected by the neighbouring APs with the overlapped effective regions. This is because the distant APs cause little gain/interference. We plot the relationship between APs and their overlapped effective region as Fig.3. Here, for notation simplicity, we only show the case where the overlapped region is affected by at most two APs. For a set of AP $\mathcal{B} = \{b, b'\}$, the state of bth AP's effective region is split into the regions with and without overlapping with neighboring b'th AP's effective region, denoted as $\overline{s}_b$ and $s_{b,b'}$, respectively ($s_b = \overline{s}_b \cup s_{b,b'}$). Then, we can write the local QoS function of b-th AP with the users inside its effective region as

$$r_t^b = r^b(\overline{s}_b, a_b) + \sum_{b' \in \mathcal{B}^{-b}} r^b(s_{b,b'}, a_b, a_{b'}) = \sum_{u \in \mathcal{U}^b} r_t^u, \tag{4}$$

where $\mathcal{B}^{-b}$ denotes the set of APs with overlapped effective region, for Fig. 3 $\mathcal{B}^{-b} = \{b'\}$, r_t^b is the local QoS function of the AP b in $\mathcal{B}$, $\overline{s}_b$ is the state information near the AP b without overlapping effective region, $s_{b,b'}$ presents the state of users in overlapped effective region, whose QoS is affected by APs from both sides. Then it is clear to see that r^b contains two parts, one refers to the dominated status $\overline{s}_b$ and another refers to the overlapped status $s_{b,b'}$.

By separating effective regions, our considered optimization problem can be decomposed into sub-problems from the view of individual APs, which optimizes their local QoS function by interacting with their opponents and the environment. The optimization target of a single agent is presented as

$$\begin{aligned} &\max_{\pi} \sum_{t}^{\infty} [\sum_{u \in \mathcal{U}^b} r_t^u | \{\mathcal{B}^u\}_{u \in \mathcal{U}} \sim (\pi_b, \{\pi_{b'}\}_{b' \in \mathcal{B}^{-b}})], \\ &s.t. \quad |\mathcal{B}^u| \leq B_{\max}, \end{aligned} \tag{5}$$

where the constrain of $\mathcal{B}^u|$ is realised by blocking certain coordination decisions via topology design. Interestingly, a similar idea of sharing partial input parameters in value function separation is proved to be effective by value decomposition network [14]. This forms our basic idea to deal with the high-dimensional large-scale communication environment.

III. Multi-Agent Reinforcement Learning Design With Problem Decomposition

Leveraging the aforementioned properties, we introduce a multi-agent RL for a large-scale multi-cell wireless network. It allows APs to optimize their long-term performance by interacting with other agents and environment [25], thus reducing the complexity.

A. Partially Observable Networked Stochastic Game Definition

To solve our considered problem with RL methods, we first show that our considered problem is Markov and define our problem as a networked partially observable stochastic game (ND-POSG) with APs as agents. The problem is Markov as the next state can be fully decided by the previous state and action. The impact of agents on neighbours can be seen as network connected. And the agent cannot fully observe the environment. Thus, we can define our considered ND-POSG via a tuple of $< \mathcal{S}, \mathcal{B}, \{\mathcal{O}^b\}, P, \{\mathcal{A}\}, \{\mathcal{A}^b\}, \{\mathcal{R}^b\}, \Omega >$. We define each component of this tuple notations as

- $\mathcal{S}$ is a set of joint state ($s \in \mathcal{S}$), and $\mathcal{S}^b$ represents the set of local state of agent b ($s_b \in \mathcal{S}^b$). State s_b contains all information of the system inside the effective region of b-th AP, which includes users' position, SINR, neighbouring AP cooperation state, and AP's transmit power, etc.

- $\mathcal{B}$ is the set of agents ($b \in \mathcal{B}$), which are co-located with each AP.
- $\mathcal{O}^b$ is a set of local observations of the b-th agent ($o_b \in \mathcal{O}^b$). The observation presents the accessible information in the state by APs, which contains users' location, users' QoS, and neighbouring AP's location in our considered APs. Our considered problem is partially observable in this case.
- P is transition probability which denotes the transition probability from any state-action pair to the next state, i.e. $P(s'|s,a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S}$,
- Ω is the probability to generate o_b from s_b for agent b, i.e. $\Omega(o_b|s_b) : \mathcal{S}^b \times \Omega$. The physical meaning of Ω is the accessibility of information in the state, decided by APs' sensors' capability.
- $\mathcal{A}$ is the set of joint actions of all APs. The local action set of the bth AP is given as $\mathcal{A}^b$ ($a^b \in \mathcal{A}^b$). The action is the cooperation request defined before. In our considered scenario, the size of local action space is defined as $|\mathcal{A}^b| = \sum_{c=0}^{C} |\mathcal{B}^{-b}|!/(c!(|\mathcal{B}^{-b}| - c)!)$, where C is the maximum group size to enforce the constrain in Eq.(2). The size of joint action space increases exponentially with the number of cooperative APs and the maximum group size, i.e. $|\mathcal{A}| = (|\mathcal{A}^b|^{|\mathcal{B}|})$, which causes the scalability problem.
- The optimization target r^b (given in Eq.(4)) is used as the reward function for the bth agent, i.e. $r^b(s,a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. r is the overall sum reward for all agents.

In each round of decision, each AP observes its surrounding environment from state S_t with Ω. Then, each AP chooses its action according to its local policy π_{θ^b}, which gives the probability of choosing action a^b with observation o_b, i.e. $\pi_{\theta^b} : \mathcal{O}^b \times \mathcal{A}^b \rightarrow [0,1]$. The policy is defined by a set of parameters θ^b. Then, the cooperation groups throughout the network are formed, i.e. A_t through hand-shake.

The network then serves the users with the current cooperation groups for a certain time slot, each AP observes a local reward, which shows the efficiency of the current cooperation decision. The reward is then used to update the local policy. The network shifts to a new state S_{t+1}, while all APs observe their observation O^b_{t+1}. The aforementioned process is repeated.

To conclude the properties mentioned in the aforementioned ND-POSG and simplify further discussions, we make the assumption on the transition functions and policy functions of agents, which is reasonable and standard for neural network-based RL:

Assumption 1: We assume that the policy function among agents is statistically independent. Thus, the joint policy π of all agents is factorized as the product of all local policies, i.e. $\pi_\theta = \prod_{b\in\mathcal{B}} \pi_{\theta^b}(o_b, a_b)$. Also, the policy function is differentiable with respect to all possible parameters θ^b. As such, we can write the state transition probability between two state s and s' ($s, s' \in \mathcal{S}$) under a joint policy θ as

$$\mathbb{P}_\theta(s'|s) = \sum_{a\in\mathcal{A}} \prod_{b\in\mathcal{B}} \pi_{\theta_b}(o_b, a_b)\Omega(o_b|s_b)P(s'|s,a), \quad (6)$$

where $\theta = [\theta_b]_{b\in\mathcal{B}}$. The proposed Markov chain is irreducible and aperiodic under any policy set π_θ. To simplify the notations, we write $\pi_\theta(s,a) = \prod_{b\in\mathcal{B}} \pi_{\theta_b}(o_b, a_b)\Omega(o_b|s_b)$, which introduces the partially observable cases into proposed stochastic game [26].

The Markov chain is irreducible and aperiodic means that it has a stationary distribution of the existence of state s under the policy defined by θ, which is denoted as $d_\theta(s)$ for any s. These assumptions are critical for methods like policy gradient and are satisfied by policies defined by neural network parameters. Then, we write our long-term optimization goal for the considered stochastic game by introducing joint policy π_θ and rewriting the QoS in Eq.(2) with state and actions defined in ND-POSG

$$\max_\theta J(\theta) = \mathbb{E}_{(s,a)\sim\mathbb{P}_\theta(s,a)}[\sum_{t=0}^{\infty} r(s,a)],$$
$$s.t. |\mathcal{B}^u| < B_{\max} \quad (7)$$

where $P_\theta(s,a) =$ is the probability of the state-action pair (s,a).

B. Partial Derivation of Policy Gradient Method

In this section, we apply the policy gradient methods to solve our considered optimization problem and present the gradient update steps. The optimization target is to find a θ^* which maximizes the target function $\theta^* = \arg\max_\theta J(\theta)$. The optimization algorithm steps θ in the direction of the gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_{(s,a)\sim\mathbb{P}_\theta(s,a)}[Q_\theta(s,a)\nabla_\theta \log \pi_\theta(a|s)]. \quad (8)$$

We consider the Q-function as the expected reward of the overall possible state-action pair under the policy π_θ, i.e. $\mathbb{P}_\theta(s,a)$, which is different from the episodic case of RL where the environment can be reset after certain steps. It allows the RL to learn from limited initial state distribution, which eases the learning steps [20]. However, in our considered scenarios, such a reset is impossible in a real-world deployment. This is the so called non-episodic reinforcement learning, that is commonly found in wireless networks, but rarely studied.

Clearly shown in Eq.(8), the key point lies in precise estimation of Q_θ. To enable distributively training, we try to decompose the problem of updating θ parameters into local updates of the individual agent, while still solving the global optimization problem. We first formulate the local update for θ_b at each agent by taking partial derivation of the original target function for the local parameter θ_b. Then, we have

$$\nabla_{\theta_b} \log \pi_\theta(a|s) \overset{(a)}{=} \nabla_{\theta_b} \log \pi_{\theta_b}(a_b|o_b), \quad (9)$$

in (a) we apply the assumption of independent local policy in the Assumption.1, and $\nabla_{\theta_b} \log \Omega(o_b|s_b) = 0$ following the definition that observation function does not correlates with θ. We then take the partial derivation of target function $J(\theta)$ in Eq. (10). The function can be simplified according to Assumption 1.

$$\nabla_{\theta_b} J(\theta) = \mathbb{E}_{(s,a)\sim\mathbb{P}_\theta(s,a)}[Q_\theta(s,a)\nabla_{\theta_b} \log \pi_{\theta_b}(a_b|o_b)]]. \quad (10)$$

The result of Eq.(10) shows that the update in local parameter sets θ_b can still optimize the overall optimization problem if

we can obtain a correct estimation of the global Q-function $Q_\theta(s,a)$.

C. Decomposition in Q-Function Estimation

In the following, we propose the idea of Q-function decomposition to get a local approximation of $Q_\theta(s,a)$ by considering several unique characteristics in wireless networks. We first recall that the agents have a limited effective region of the nature of the wireless signal. Thus, we can rewrite the global Q-function in Eq.(8) by keeping the related part in the global Q-function described by agent b's effective region

$$Q_{\theta_b}(s,a) \approx Q_{\theta_b}(s_b, a_b, \{a_{b'}\}_{b'\in\mathcal{B}^{-b}}) \tag{11}$$

where $\mathcal{B}^{-b}$ is the set of neighbouring agents of b, which have overlapped effective regions with agent b. From the Eq.(11), we can see the local update can still optimise the global target with known neighbouring actions in the Q-function estimation. Although the required information is largely reduced, action sharing can be hard as the neighbouring agents with the overlapped effective region are hard to identify and can be outside the communication range.

Without the information from neighbouring agents, we need to quantify the influences. We recap the properties that the Q-function is geometrically in-correlated and independent (QoS function in Eq.(1)). Considering the effective region, as shown in Fig.3, we can rewrite the local Q-function into overlapped and non-overlapped effective regions as

$$\begin{aligned} &Q_{\theta_b}(s_b, a_b, \{a_{b'}\}_{b'\in\mathcal{B}^{-b}}) \\ &\quad= Q_{\theta_b}(\overline{s}_b, a_b) + \sum_{b'\in\mathcal{B}^{-b}} Q_{\theta_b}(s_{b,b'}, a_b, a_{b'}) \\ &\quad\overset{(a)}{=} \mathbb{E}_{a'_b\sim\pi_{\theta^b}(a'_b|s'_b)}\Big[\sum_{t'=t}^{\infty}[r(\overline{s}'_b, a'_b) \\ &\qquad+ \sum_{b'\in\mathcal{B}^{-b}} \mathbb{E}_{a'_{b'}\sim\pi_{\theta^{b'}}(a'_{b'}|s'_{b'})} r(s'_{b,b'}, a'_b, a'_{b'})]\Big], \end{aligned} \tag{12}$$

where through (a), we expand the Q-function based on the definition of Q-function.

The Q-function is decomposed into stationary term referenced on local information ($\overline{s}_b$ and a_b) and cooperation term referenced on neighbors' information ($s_{b,b'}$ and both a_b and $a_{b'}$). The stationary term is fully dominated by the current agent with negligible influences from other agents. Thus, the precise estimation can be easily and precisely obtained for $Q_{\theta^b}(\overline{s}_b, a_b)$. The cooperate term is non-stationary without information of neighbour agents' actions or policies. Such a non-stationary environment is the major reason limiting the performance of our considered multi-agent system. Note that the change of neighbours' actions/policies can easily make previous experience expire, as the optimal policy is referenced on neighbours' policies. The information difference can be quantified by the estimation error of the Q-function. The estimation error $\epsilon(\omega^b)$ caused by a non-stationary environment can be written as

$$\epsilon(\omega^b) = Q_{\omega^b}(s_{b,b'}, a_b) - \mathbb{E}_{a_{b'}\sim\pi_{\theta^{b'}}(a_{b'}|s_{b'})} Q_{\theta^b}(s_{b,b'}, a_b, a_{b'}) \tag{13}$$

where ω^b is the parameter of the estimator in b-th agent for Q-function.

D. Parameter Sharing for Non-Stationary Environment

Sharing the information of actions and policies among neighbours is a potential solution to enhance the performance of the multi-agent algorithm in a non-stationary environment. In this section, we introduce the idea of applying parameter sharing among all agents, where the features are naturally shared via aligned global models and commonly observed environment features without frequent communication.

There are two major benefits of applying parameter sharing for a non-stationary environment: convergence speed and features sharing. The benefit of convergence speed in applying parameter sharing is well-proof by CTDE or federated learning [5], [8]. The sharing of the knowledge learnt from the environment among the network boosts the individuals' learning process, which supports the fast adjustment to the new optimal policy. Especially with certain homogeneous agents, i.e., users' locations follow PPP distribution, the knowledge of certain agents can nearly fully immigrate among the network. Among the parameter sharing methods, federated averaging (FedAvg) is a commonly used method, where the distributed model can be trained using local data captured by each agent and then averaged and aggregated as a global model.

The estimation error caused by unknown neighbours' actions can also be reduced by parameter sharing. During the parameter sharing process, a combined global policy is aggregated from and shared with all agents. With aligned policy π_θ known for all agents, the estimation error in the non-stationary term of Eq. (13) (due to the unknown neighbours' policy) is reduced to

$$\epsilon(\omega^b) = Q_\omega(s_{b,b'}, a_b) - \mathbb{E}_{a_{b'}\sim\pi_\theta(a_{b'}|s_{b'})} Q_\theta(s_{b,b'}, a_b, a_{b'}), \tag{14}$$

where $\theta = 1/|\mathcal{B}|\sum_{b\in\mathcal{B}}\theta^b$ and $\omega = 1/|\mathcal{B}|\sum_{b\in\mathcal{B}}\omega^b$ are the global aggregated parameters for policy and Q-function if considering federated average (FedAvg) algorithm, which is a popular parameter sharing algorithm [27]. With shared parameters in Q-function estimation and policy, neighbouring agents can cooperate effectively through observed features from the common observable overlapped effective region naturally, i.e. $s_{b,b'}$ in Fig.3, without communication, as the Q-function references on agent's local features and neighbours' actions. In this way, the consistency can be maintained among agents and estimation error is reduced. However, the non-stationary problem is still not fully solved and there is no guaranteed optimality of our proposed algorithm, as the remaining estimation error in Eq.(14) still influenced by the unknown state in neighbour's observation:

$$\begin{aligned} \overline{\epsilon}(\omega^b) = \mathbb{E}_{a_{b'}\sim\pi_\theta(a_{b'}|s_{b,b'})} Q_\theta(s_{b,b'}, a_b, a_{b'}) \\ - \mathbb{E}_{a_{b'}\sim\pi_\theta(a_{b'}|s_{b'})} Q_\theta(s_{b,b'}, a_b, a_{b'}). \end{aligned} \tag{15}$$

We also notice that such estimation error decreases with an increasing portion of the overlapped effective region $s_{b,b'}$ (allowing more features to be shared), which enhances the performance of cooperation. Meanwhile, when the portion

Algorithm 1: Parameter Sharing Reinforcement Learning Algorithm for ND-POSG

Initiate environment Env, state s_0, and the initial values of the parameters $\{\theta^b\}_{b\in\mathcal{B}}$ and $\{\omega^b\}_{b\in\mathcal{B}}$.
repeat
  if `Game end` **then**
    Reset Env and $t=0$, obtain new S_0
  for $i \in \mathcal{B}$ **do**
    Obtain O_t^b from S_t
    Select an action $A_t^b \sim \pi_{\theta_t^b}(O_t^b)$
  Forms joint action $a_t = (A_t^b)_{b\in\mathcal{B}}$, the environment move to S_{t+1}
  for $i \in \mathcal{B}$ **do**
    Observe local reward r_t^b from S_{t+1}
    Update actor's and critic's parameters following Eq.(18) and Eq.(16) or categorical algorithm and the error from CORAL
  Update global model by averaging θ_t^b and ω_t^b
  Update average reward following (17)
until *Performance Not Improved*

of the overlapped effective region is small, the error is also small and tolerable. Interestingly, the variance of $\mathbb{E}_{a_{b'}\sim\pi_\theta}$ decreases with the increasing number of overlapping agents. This reduces the effectiveness of our considered algorithm with a high degree of connected APs and is the remaining problem of our proposed multi-agent algorithm. The problem can be partly solved by applying distributional estimation with the cost of complexity [28].

IV. Algorithm Design

We present our proposed parameter sharing RL algorithm for the considered ND-POSG in the multi-cell network. This framework can be implemented with various RL approaches. Here, we show the algorithm with a policy-based actor-critic RL algorithm, where the Q-function is estimated in the critic part to guide the update of the policy generated by the actor part.

As defined before, the parameters of policy and Q-function estimator in agent b at time t is denoted as θ_t^b and ω_t^b. The critic, i.e. Q-function estimator, should tell whether the current policy is better than the average reward. The average reward is calculated via experience and given as $r(\pi_\theta) = \mathbb{E}_{(s,a)\sim\mathbb{P}_\theta(s,a)}[r(s,a)] = \sum_s d_\theta(s)\sum_a \pi_\theta(a|s)r(s,a)$. With local state S_t^b and action A_t^b, the network update with TD-error and estimated average reward $\hat{r}_t^b$ at b-th agent follows

$$\overline{\omega}_{t+1}^b \leftarrow \omega_t^b + \alpha_t^\omega \nabla_\omega Q_{\omega_t^b}(S_t^b, A_t^b, A_t^{-b})(r_{t+1}^b - \hat{r}_t^b + Q_{\omega_t^b}(S_{t+1}^b, A_{t+1}^b, A_t^{-b}) - Q_{\omega_t^b}(S_t^b, A_t^b, A_t^{-b})), \quad (16)$$

where α_t^ω is the step size for the critic network. With distributional RL, the TD error is the cross-entropy loss of the KL divergence between the current return and the estimated distribution of the return following the categorical algorithm in [28, Algorithm. 1]. The average reward is updated via

$$\hat{r}_{t+1}^b \leftarrow \hat{r}_t^b + \alpha^r(r_{t+1}^b - \hat{r}_t^b + Q_{\omega_t^b}(S_{t+1}^b, A_{t+1}^b, A_t^{-b}) - Q_{\omega_t^b}(S_t^b, A_t^b, A_t^{-b})), \quad (17)$$

where α^r is the reward update parameter. Guided by the critic, the actor is updated as follows

$$\overline{\theta}_{t+1}^b = \theta_t^b + \alpha_t^\theta \nabla_\theta \log \pi_{\theta^b}(O_t^b|S_t^b) Q_{\theta_t^b}(S_{t+1}^b, A_{t+1}^b, A_t^{-b}), \quad (18)$$

where α_t^θ is the step size for the critic network. Then, in each parameter sharing step, there are two sharing methods, federated average (FedAvg) and federated transfer learning (FedTrans) with CORelation ALignment (CORAL). With FedAvg, the parameter of actor and critic is simply aggregated and averaged

$$(\omega_{t+1}, \theta_{t+1}, \hat{r}_{t+1}) = 1/|\mathcal{B}| \sum_{b\in\mathcal{B}} (\overline{\omega}_t^b, \overline{\theta}_t^b, \hat{r}_{t+1}^b) \quad (19)$$

With FedTrans, CORAL is a simple approach for unsupervised model alignment, which minimizes the model shift between the global model and local model [29]. The difference between these two aggregation methods is discussed later.

A. Converge Condition

The convergence for our considered fully cooperative multi-agent problem is promising. We first make assumptions on the update rate of ω and θ, which is common in RL convergence proof.

Assumption 2: The update rate of α^ω and α^θ satisfy

$$\sum_t \alpha_t^\omega = \sum_t \alpha_t^\theta = \infty, \ \sum_t (\alpha_t^\omega)^2 + \sum_t (\alpha_t^\theta)^2 \le \infty$$

Then, following Kushner-Clark Lemma, the convergence can be proved with four conditions for constructing ordinary differential equations. Before continuing the proof, we first need to convert the update process in Eq.16 into metric form for presentation convenience. We make several basic assumptions, which align with the neural network properties. The Q-function estimator generates its results reference on ω, local features from states, and actions.

Assumption 3: For the agent b, the Q-function can be written as a combination of features from independent locations: $Q^b(s_b, a^b, a^{-b}) = \omega^\top \phi(s_b, a^b, a^{-b})$, where $\phi(s_b, a^b, a^{-b}) = [\phi^1(s_b, a^b, a^{-b}), \dots, \phi^K(s_b, a^b, a^{-b})]^\top \in \mathbb{R}^K$. The feature matrix $\Phi = [\phi(s_b, a^b, a^{-b})] \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}^b||\mathcal{A}^{-b}|\times K}$ is uniformly bounded and full rank, which means 0 is not a eigenvalue of Φ (there does not exist a vector $v \in \mathbb{R}^K$ which gives $\phi v = \mathbb{1}$).

Then, to show the convergence of our proposed parameter sharing algorithm, we first analyze the critic step's convergence while assuming a fixed policy π_θ following the two-time-scale SA analysis [30]. The convergence of actor step upon converged critic is nicely shown by literature. First, it is possible to consider the critic update via an ordinary differential equation

$$\dot{z} = \Phi^\top D_\theta^{s,a} r^b(s,a) - \Phi^\top D_\theta^{s,a}(P^\theta - I)\Phi\omega, \quad (20)$$

where $D_\theta^{s,a}$ is the probability of the existence of state-action pair (s,a), i.e. $D_\theta^{s,a} = \text{diag}[d_\theta(s)\pi_\theta(a|s), s \in \mathcal{S}, a \in \mathcal{A}]$, P^θ is the transition probability from (s,a) to (s',a') under policy θ, i.e. $P^\theta(s',a'|s,a) = P(s'|s,a)\pi_\theta(s',a')$.

We now justify why the update steps satisfy the Assumptions.1-4 for Kushner-Clark Lemma [31]: 1) Since the $\frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}}\delta_t^b$ is the function of ω_t^b, i.e. $\delta_t^b = r_{t+1}^b + (\phi_{t+1}^b)^\top\omega_t - (\phi_t^b)^\top\omega_t$. Then, with uniformly bounded ϕ, δ is Lipschitz continuous in ω_t, as all components are linear; 2) P^θ is a non-negative matrix. According to the Perron-Frobenius theorem, P^θ has one eigenvalue equal to the spectral radius of P^θ, whose maximum value is 1 for the considered probability transfer matrix. Other eigenvalues are less than 1. Thus, it is possible to have an zero eigenvalue in vector $P^\theta - I$, which gives a vector v that satisfies $\Phi v = \mathbb{1}$. However, this special case rarely exists. All eigenvalues are the negative real number in $P^\theta - I$. Hence, Eq.(20) has a asymptotically stable solution (equilibrium) [31, Theorem. 2] when

$$\Phi^\top D_\theta^{s,a}[r^b(s,a) - (P^\theta - I)\Phi\omega] = 0, \tag{21}$$

where the solution ω_θ is unique; 3) The step size α_t^ω has the property in Assumption.2; 4) The FedAvg operation keeps local parameters aligned with global parameters. Thus, this condition is absent. In this way, the update of the critic part follows the Kushner-Clark Lemma, which converges almost surely when $t \to \infty$. Thus, we complete the proof of the critic convergence [26]. Then, following the proof of the original actor-critic algorithm and two-time-scale SA analysis [32], the actor part can converge guided by a converged critic, which concludes the proof.

B. Convergence Speed Analysis With Informational Model

Normally in our considered FedAvg parameter sharing method, the averaging operation is required to be performed in every learning step. It is resource and time-consuming to transmit the entire model for a round-trip each time, which is also unrealistic for a specific communication system. But reducing the parameter sharing frequency can significantly weaken the benefit of parameter sharing. Thus, it is necessary to analyze the effect of parameter sharing frequency on the convergence speed of our proposed multi-agent algorithm. In this section, we applied the informational model for multi-agent learning defined in [33] and extended it to FedAvg cases to derive the upper bound for converging speed under different parameter sharing frequencies for our model.

Similar to Eq.(12), we can separate the knowledge or information required to fit the Q-function at each agent b into local information (information in $\overline{s}^b$) and cooperative information (information in $s^{b,b'}\forall b' \in I^{-b}$). We also define the local information in b-th agent at time t as $\mathcal{I}_{b,\text{env}}(t)$ and the cooperative information between b and its neighbor b' as $\mathcal{I}_{b,b'}(t)$. In this way, we have the overall information in bth agent at time t as

$$\mathcal{I}_b(t) = \mathcal{I}_{b,\text{env}}(t) + \sum_{b'\in\mathcal{B}^{-b}} \mathcal{I}_{b,b'}(t). \tag{22}$$

During the learning procedure, the information increases over each time step. For any agent, b in a group of agent $\mathcal{B}$ with neighbour agents $\mathcal{B}^{-b}$, the information gained in each learning time step is defined as

$$\Delta^\uparrow\mathcal{I}_b(t) = \Delta^\uparrow\mathcal{I}_{b,\text{env}}(t) + \sum_{b'\in\mathcal{B}^{-b}} \Delta^\uparrow\mathcal{I}_{b,b'}(t), \tag{23}$$

where $\Delta^\uparrow\mathcal{I}_{b,\text{env}}$ is the gain for local information, $\Delta^\uparrow\mathcal{I}_{b,b'}$ is the gain for cooperation information between agent b and its neighbor b'. For any agent b in a group of agent b with neighbor agents I^{-b}, the local information required to converge is defined as $\mathcal{C}_{b,\text{env}}$ and the cooperative information between b and neighbor b' is $\mathcal{C}_{b,b'}\forall b' \in I^{-b}$. These two terms satisfy

$$\mathcal{C}_{b,\text{env}} + \sum_{b'\in I^{-b}} \mathcal{C}_{b,b'} = 1, \mathcal{C}_{b,\text{env}} \in [0,1], \mathcal{C}_{b,b'} \in [0,1]. \tag{24}$$

To model the value of information gain, we denote the function of the information gain learnt as Λ, which is a function of the rest of the information. Then, the information gained for local information and cooperation information can be written as

$$\Delta^\uparrow\mathcal{I}_{b,\text{env}}(t) = \mathcal{K}_{b,\text{env}}\Lambda\left(\mathcal{C}_{b,\text{env}} - \mathcal{I}_{b,\text{env}}(t-1)\right), \text{ and} \tag{25}$$

$$\Delta^\uparrow\mathcal{I}_{b,b'}(t) = \mathcal{K}_{b,b'}\Lambda\left(\mathcal{C}_{b,b'} - \mathcal{I}_{b,b'}(t-1)\right), \tag{26}$$

respectively, where the value of $\mathcal{K}_{b,\text{env}} \in [0,1]$ and $\mathcal{K}_{b,b'} \in [0,1]$ refer to the learning rate coefficient, which corresponds to the several settings in the algorithm, such as batch size, learning rate, etc, and may differ among agents. It should be noted that the learning function has the property as $\Lambda(x) \leq x$, since the learnt information can't be larger than the rest unlearnt information.

As illustrated, the change of neighbours' policy can make the previously learnt information outdated. The information loss is highly correlated to the amount of new information learnt by neighbour agents, which is unknown to the current agent. As the local information, $\overline{s}^b$ can be seen as stationary so there is no information loss in the learning the local information part. For the cooperation information, we define the information loss between agent b and b' as

$$\Delta^\downarrow\mathcal{I}_{b,b'}(t) = \frac{\Delta^\uparrow\mathcal{I}_{b'}(t)}{\mathcal{I}_{b'}(t-1) + \Delta^\uparrow\mathcal{I}_{b'}(t)}\mathcal{I}_{b,b'}(t-1). \tag{27}$$

With the help of parameter sharing, agents can share the information learnt locally among the group of agents. Moreover, after the agents share the same learning model after FedAvg, it has full information for the neighbour agents in the next learning step. In this way, there is no information loss after each FedAvg operation. Following these definitions and combining the gain and loss of information in each update, we solve t for the upper bound t when $\mathcal{I}_{b,b'}(t) \leq \mathcal{C}_{b,b'}(1-\epsilon)$ or generally $\mathcal{I}_{*,*}(t) \leq \mathcal{C}_*(1-\epsilon)$ for convergence of neighbouring part as Eq.(28), shown at the bottom of the next page. The detailed proof is listed in Appendix.

The upper bound of converge time should be the maximum number within the upper bound of cooperation information and local information. Here, we only visualized the upper

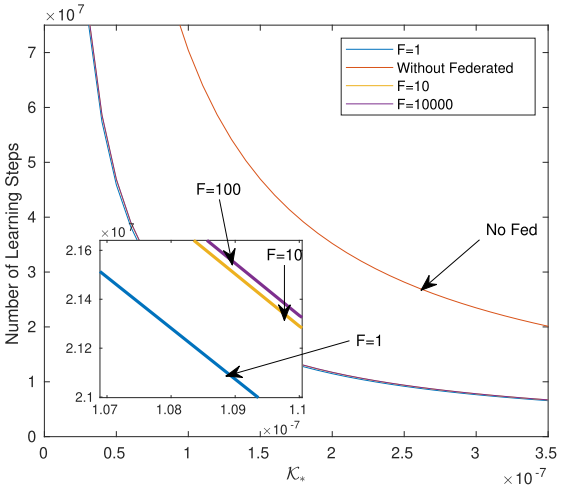

Fig. 4. Converge Rate Over $\mathcal{K}_*$ with Different F.

bound of cooperation information and its relations between F, as the local information is investigated in [33]. We follow the parameter setting with the $\mathcal{C}_{*,\text{env}} = 0.1$, $|\mathcal{B}| = 10$, $\mathcal{I}_{*,*}(0) = 0.01$, and $\epsilon = 0.001$ in [33]. We plot the Fig.4 for Eq.(28) which shows that the high FedAvg frequency can significantly reduce the required learning steps, and FedAvg can still significantly accelerate convergence even with relative large F.

C. Centralized-Decentralized Mismatch

We motivate the use of parameter sharing in the multi-agent RL with the assumption of homogeneous devices, where the environment around agents is assumed to be similar. However, due to the different geometry characteristics in the actual deployed environment, the agents can have completely different environmental characteristics. The "averaged" model is surely not generalized enough to handle the heterogeneity environment. The sub-optimality or local characteristic of a certain agent's policy can propagate through the FedAvg process and negatively affect other agents' performance, called "Centralized-Decentalized Mismatch" [19]. In our considered CoMP case, those APs located near field edge or obstacles which limit the visit of users, learn highly personalized action policy. Such policy not suitable to be fully shared and accepted by other agents. In reality, users are not distributed evenly in the serving range of each agent while being served due to the existence of hot spots (distributed following PCP instead of PPP). Thus, it is important to balance the local knowledge and shared knowledge with the case of centralized-decentralized-mismatch.

The centralized-decentralized-mismatch problem can be potentially solved by applying transfer learning methods rather than directly averaging [34]. With transfer learning, it is possible to guide the local update with the global model

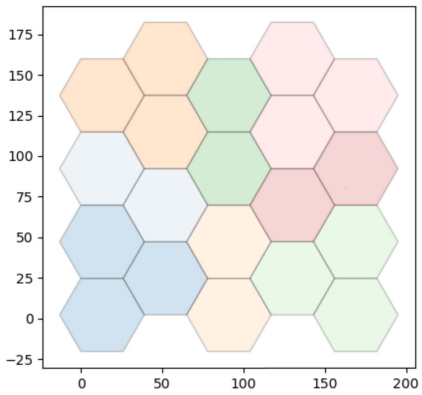

Fig. 5. Fixed cooperation scheme for 20 APs or the learning policy under PPP user distribution.

while keeping the local characteristics and reducing the negative effect of centralized-decentralized mismatch. We adopt a transfer learning method CORAL to minimise the difference between the local model and the global model, called FedTrans. CORAL is a simple approach for unsupervised model alignment, which minimizes the model shift between the global model and local model [35]. By minimizing the second-order statistics between local and global features, the CORAL helps the global knowledge to be transferred into the local model while keeping the characteristic of local features. We apply CORAL to our algorithm where the loss of CORAL is measured between the output of linear layers before the softmax layers [29].

V. Simulation Results

In this section, we provide simulation results to show the effectiveness and scalability of our proposed parameter-sharing multi-agent RL frameworks and verify several algorithm designs with our simulated CoMP environment.

We consider a $182m \times 168m$ serving area with $|\mathcal{U}| = 160$ PPP/PCP distributed users in 10 groups with largest radius of 40 m. Users' locations follow PCP distribution with 10 groups in the area. The new users' position is generated every time slot and waits for two-time slots before the service fails. The users are served by $|\mathcal{B}| = 5 \times 4 = 20$ APs. The APs are distributed in cellular with 6 neighbours. The gap between neighbouring APs is $44.3m$ or $52m$. The number of possible cooperation actions is $|\mathcal{A} = 12|$, as the maximum size of the group is considered as 3. The APs choose one or two of its neighbour to cooperate with.

For the baseline, we use the fixed cooperation scheme. As the cooperation can only gain system performance, thus, the greedy scheme will lead to a fixed cooperation scheme with maximum cooperating APs, which can be seen as a local optimum. The resulting cooperation groups are presented in Fig.5, where the APs with the same colour are cooperating. For

$$t^* = \frac{F \log \frac{\mathcal{C}_* \epsilon}{\mathcal{C}_* - \mathcal{I}_{*,*}(0)}}{\log\left[1 - |\mathcal{B}|\left[(1-\mathcal{K}_*)(1-(1-\mathcal{K}_* \quad (1 - \frac{\mathcal{I}_{*,*}(0)}{\mathcal{C}_{*,\text{env}}/|\mathcal{B}^b| + \mathcal{K}_*\mathcal{C}_* + \mathcal{C}_*}))^{F-1}) + \mathcal{K}_*\right]\right]} \tag{28}$$

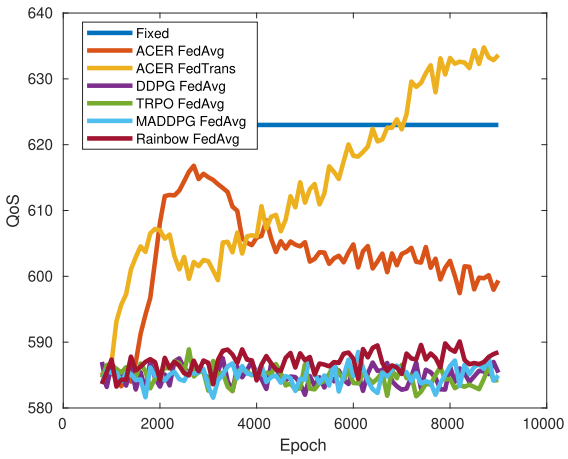

Fig. 6. QoS Performance of different RL algorithms under our introduced multi-agent algorithm with FedAvg.

the interaction between environment and learning algorithm, all agents learn from the environment continually without resetting to some default states, i.e., non-episodic. Empirically, the non-episodic environment increases the learning difficulty. The detailed discussion is out of the scope of this paper [20].

For the neural network design, we adopt a three convolutional layer neural network to capture the geometry correlation between APs and users. The network takes a picture whose value of the pixel presents the existence of neighbour APs or users in the corresponding location. The value is added and normalized if multiple users are located in the same pixel. In this way, the network takes the users' geometry information with constant input size. To capture the time-correlate information, three nearby captures are combined and fed into the network. We directly feed multiple observations into the network without a recurrent network as the requests expire within several time instants. After the input is processed as a hidden vector by the convolutional layer. The hidden vector then is then used to generate the Q-function estimation by a distributional RL structure with three noisy linear layers [28]. The noisy linear layers add noise to the result for state-based exploration [36]. The distributional RL structure allows the estimation to be performed on the value following certain distribution precisely, which is suitable for wireless communication cases. The CORAL layer is added at the end of the network to perform transfer learning.

For the detailed RL algorithms, we show the performance of different RL approaches with a federated average algorithm in an CoMP environment. All these approaches are applied individually at each agent with parameters averaged and shared among agents. In Fig.6, we consider actor-critic with experience replay with FedAvg or transfer learning method (ACER FedAvg/FedTrans) with value function in critic, Deep Deterministic Policy Gradient (DDPG) [37], Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [9], Trust Region Policy Optimization (TRPO) [15], [38], and Rainbow [36].[1] We do not test some multi-agent algorithms in the literature,

[1]The authors acknowledge the use of the research computing facility at King's College London, Rosalind (https://rosalind.kcl.ac.uk). The code for this paper is available in https://github.com/paperflight/Fed-MF-MAL/tree/main.

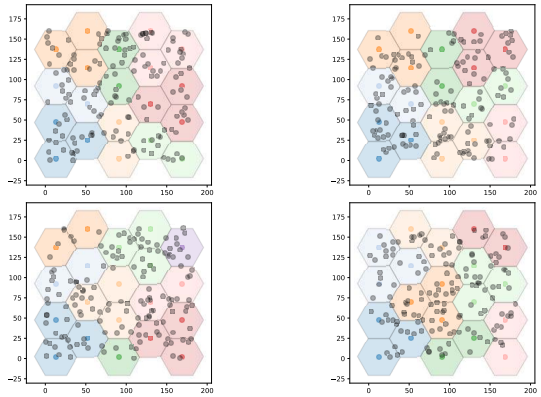

Fig. 7. Several examples of grouping decisions made by ACER FedTrans algorithm, where the black spots present users.

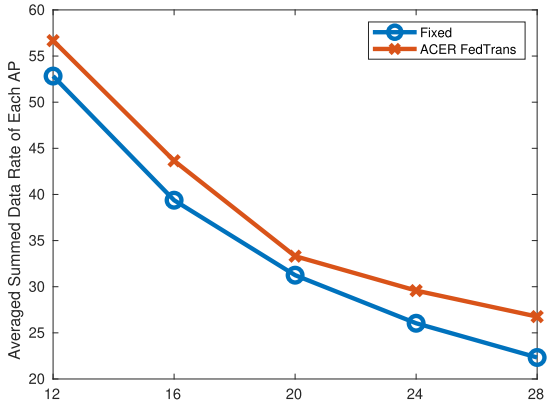

Fig. 8. Averaged summed data rate with different number of APs using fixed or ACER FedTrans method.

for example, QMIX, as they still require centralized critic, which is nearly impossible to be trained in our scenario. All methods share a similar size of neural network with similar computation complexity. Our result shows that only ACER algorithms can achieve acceptable performance, due to the gradient update of ACER algorithm has sufficient theoretical proof for fully decentralized scenarios [26]. We apply ACER FedAvg/FedTrans algorithm for later analysis. We show some cooperation decisions made by ACER FedTrans in Fig.7.

A. Complexity Analysis

The size of this association problem in our defined environment is 12^{20}, which is over 4×10^{22} and impossible to be solved without a supercomputer. We use two cores with 3.0 GHz clock rate for each agent. Thus, the scalability of our introduced multi-agent algorithm can already be proved by this parameter setting. As shown in Fig.8, we plot the averaged summed data rate per AP with different numbers of AP. Our results show its efficiency with different scales of networks and keep outperforming the fixed scheme. By leveraging the advantage of the communication environment, the system complexity is degraded from $\mathcal{O}(|\mathcal{A}|^{|\mathcal{B}|})$ to $\sum_{|\mathcal{B}|}\mathcal{O}(|\mathcal{A}|)$ and can be applied in computation resource and memory limited devices. These designs significantly reduce the complexity of cooperative algorithms in large-scale networks.

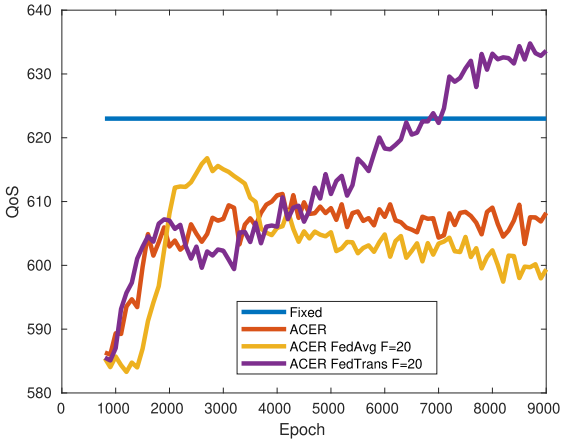

Fig. 9. QoS Performance of ACER algorithm with FedAvg algorithm or FedTrans under our introduced multi-agent algorithm.

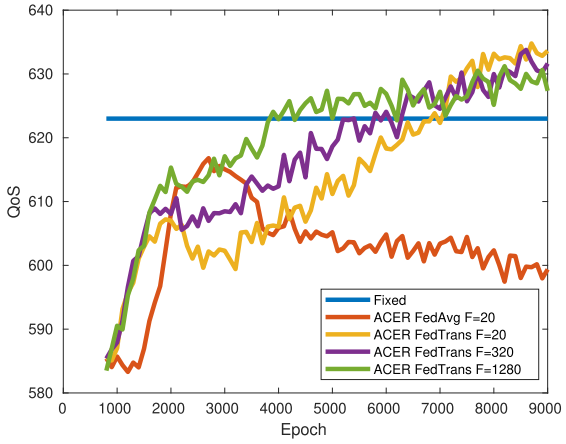

Fig. 10. The influence of parameter sharing frequency on the performance of ACER algorithm with FedTrans.

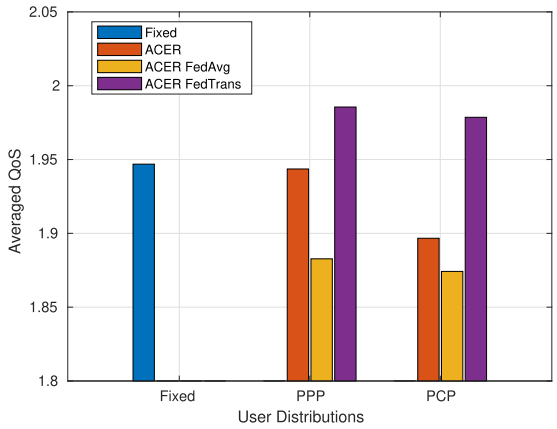

Fig. 11. The difference in averaged QoS performance for ACER algorithms with PPP distributed users or PCP distributed users.

B. Parameter Sharing via Transfer Learning

Following our analysis of how parameter sharing supports multi-agent cooperation in wireless communication scenarios, parameter sharing can stabilize the learning procedure by sharing other agents' policies in a non-stationary environment. As shown in Fig.9, parameter sharing via FedAvg can effectively improve the QoS performance. However, the ACER FedAvg removes personal characteristics at each agent, which can de-stabilise the converge process. The policy generated by ACER FedAvg tends to be a fixed scheme with zeros and infinities inside the model. After that, the QoS performance drops due to these extremely biased experiences from agents, and is even worse than the ACER. There are two reasons for this. First, the fixed scheme is a locally optimal policy. As in our case, we only have QoS performance gain for cooperation, which means that the fixed scheme maximizes the cooperating agents. This policy can be learnt without the knowledge of neighbours. The agents can always expect a cooperation hand-shake by keeping choosing certain cooperation decisions. Then, the policy is fixed to a certain action. Such knowledge is harmful and useless for the global model. Another reason is that over half of the agents (14 in 20) are located at the edge of the network in our simulation environment, whose experience highly differs from each other. The agent at the edge can only cooperate with a very limited amount of neighbours (2 in the corner and 4 at the edge). Thus, performing the FedAvg method and fully accepting this incomplete and highly personalized knowledge results in a meaningless global model. Thus, the QoS performance drops after achieving sub-optimal solutions. It highlights the necessity of employing transfer learning approaches to share the common knowledge and maintain the personality of each agent without simply averaging and replacing. As shown in Fig.9, the ACER FedTrans algorithm with CORAL effectively supports the cooperation and significantly outperforms the ACER FedAvg algorithm and Fixed scheme. The performance difference shown in the figures is limited as the cell-centre users' QoS are included while CoMP only benefit the cell-edge users' QoS.

Then, we show the influence of sharing frequency on the QoS performance, as it is critical for a communication system where communication costs matter. Following our theoretical analysis in Fig.4, the parameter sharing frequency can influence the convergence speed of the algorithm, i.e., the higher the parameter sharing frequency, the faster the convergence speed. But parameter sharing can still significantly improve the convergence speed with relatively low averaging frequency. We examine this analytical result with our simulation environment and present the result in Fig.10. At the early stage of learning ($0-2000$ epoch), the learning process with low parameter sharing frequency ($F = 320-1280$) converges faster than the early stage of the learning because it allows personal characteristics at each agent and converges to local optimal shown in Fig.5. In contrast, the frequent parameter sharing operation prevents the agents from sticking to this local optimal. In the later stage (after the 2000 epoch), the agent search to jump out of the local optimum, which requires information from other agents to further improve the QoS performance. Thus, the high parameter sharing frequency ($F = 20-160$) efficiently supports the learning process with less difference in agents' policies information, which achieves slightly higher QoS performance and converge speed than the one with a low parameter sharing frequency ($F = 320-1280$).

In Fig.11, we investigate the influence of user distribution on our algorithms. We assume the users are distributed in PCP

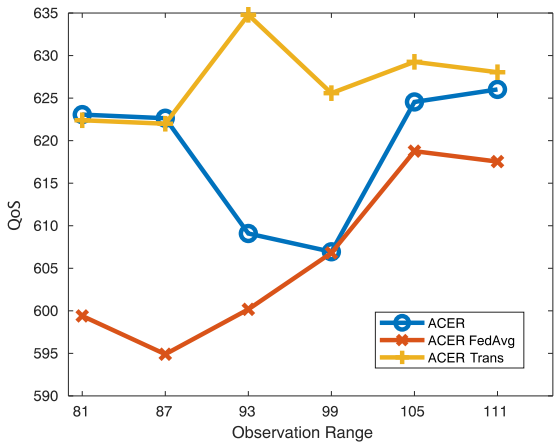

Fig. 12. The QoS performance difference in observation ranges between ACER, ACER FedAvg, and ACER FedTrans.

instead of PPP, as PCP is realistic in modelling the users' positions. PPP is easier for the multi-agent algorithm to learn since the users' positions around each agent are homogeneous, which is not the case for PCP. The PCP introduces challenges for cooperative multi-agent algorithms. As shown in Fig.11, the ACER with PPP distributed users can achieve the highest QoS performance even without parameter sharing, which is always the fixed scheme shown in Fig.5. However, the ACER performs the worse with PCP distributed users, which requires certain common knowledge between agents. The algorithms with the support of the transfer learning methods (ACER FedTrans) can obtain good QoS performance with PCP distributed users. It is also worth mentioning that we don't present the number of requests/users in the result, because the convolutional neural network handles these settings effectively and the optimal policies with different user densities are the same [18].

In Fig.12, we study the influence of the observation range on our algorithms. We pick six different observation range: 81 (39% coverage of neighbourhood's effective area), 87 (44%), 93 (49%), 99 (54%), 105 (60%), 111 (65%). The QoS performance of ACER FedAvg increases with the increasing knowledge of the neighbour's effective area, which matches our analysis. ACER fails to converge without enough knowledge from the neighbour's effective area. The ACER Trans performs well for all different observation ranges by sharing the knowledge without violating the personalities of each agent. It is worth mentioning that ACER and ACER Trans algorithm with 81 observation range generates the fixed policy in Fig.5 with little information from neighbours.

VI. Conclusion

In this paper, we introduced a parameter-sharing multi-agent RL algorithm to solve the grouping problem in coordinated multi-point communication scenarios by decomposing the optimization function geometrically. We highlighted that parameter sharing could effectively scale, accelerate the convergence speed, and enhance cooperation. We investigated the theoretical basis for the benefit of parameter sharing. We also derived an upper-bound parameter sharing multi-agent system with different parameter sharing frequencies. We have shown the necessity of using transfer learning to transfer knowledge from the global model to the local model to cope with the centralized-decentralized mismatch. We then examined our result with simulation. Our results demonstrated that our multi-agent algorithm could effectively scale and handle the problem with the relatively large amount of participating APs. We have also shown that the transfer learning methods outperform the averaging algorithm, which matches our analysis. The simulation results also show that the large observation range can help cooperation. These findings provide design insights for the future development of multi-agent algorithms in wireless communication networks.

Appendix

Combining Eq.(23) and Eq.(27), we denote the information gain $\Delta\mathcal{I}_b(t)$ for agent b from time $(t-1)$ to t as

$$\Delta\mathcal{I}_b(t) = \Delta^{\uparrow}\mathcal{I}_{b,\text{env}}(t) + \sum_{b'\in\mathcal{B}^{-b}} (\Delta^{\uparrow}\mathcal{I}_{b,b'}(t) - \Delta^{\downarrow}\mathcal{I}_{b,b'}(t)). \tag{29}$$

With the help of parameter sharing, the information gained for a local update step in the tth agent is denoted as

$$\Delta\mathcal{I}_b(t) = \Delta^{\uparrow}\mathcal{I}_{b,\text{env}}(t) + \sum_{b'\in\mathcal{B}^{-b}} (\Delta^{\uparrow}\mathcal{I}_{b,b'}(t) - \mathbb{1}[t|F]\Delta^{\downarrow}\mathcal{I}_{b,b'}(t)), \tag{30}$$

where $\mathbb{1}[t|F] = 0$, when t can be fully divided by F and the FedAvg, is performed every F local learning step.

When performing parameter sharing, the information learnt by all agents is shared and added up among agents. Thus, the information gain in FedAvg parameter sharing step after $F-1$ local update in bth agent is denoted as

$$\begin{aligned}\Delta\mathcal{I}_b(t) = \Delta^{\uparrow}\mathcal{I}_{b,\text{env}}(t) &+ \sum_{b'\in\mathcal{B}^{-b}} (\Delta^{\uparrow}\mathcal{I}_{b,b'}(t) - \mathbb{1}[t|F]\Delta^{\downarrow}\mathcal{I}_{b,b'}(t)) \\ &+ \sum_{t'=t-F+1}^{t} \sum_{b'\in\mathcal{B}/b} \Delta^{\uparrow}\mathcal{I}_{b',\text{env}}(t') \\ &+ \sum_{t'=t-F+1}^{t} \sum_{b'\in\mathcal{B}/i} \sum_{j'\in\mathcal{B}^{-b'}} (\Delta^{\uparrow}\mathcal{I}_{b',j'}(t') \\ &- \mathbb{1}[t'|F]\Delta^{\downarrow}\mathcal{I}_{b',j'}(t')).\end{aligned} \tag{31}$$

Assumption 4: To simplify the model, we assume agents are co-located in the same pattern with an identical environment. The initial amount of information in agents is the same. For simplicity, we denote the information loss between any pair of agents b and b' as $\Delta^{\downarrow}\mathcal{I}_{*,*}(t)$, i.e. $\mathcal{I}_{b,env}(t) = \mathcal{I}_{j,env}(t) = \mathcal{I}_{*,env}(t)$ and $\mathcal{I}_{b,b'}(t) = \mathcal{I}_{j,j'}(t) = \mathcal{I}_{*,*}(t), \forall i, b', j, j' \in I, i \neq j$. Besides, the information gained at each agent is also assumed to be homogeneous, which significantly reduces the complexity of our analysis. But it reduces generalization for our analysis [33].

Following Assumption.4 and Eq.(31), the overall gain in $F-1$ agent's cooperation information update and the following

parameter sharing update can be written as

$$\begin{aligned}\mathcal{I}_b(t)-\mathcal{I}_b(t-F)=&|\mathcal{B}|\sum_{t'=t-F+1}^{t}\Delta^{\uparrow}\mathcal{I}_{*,\text{env}}(t')\\&+|\mathcal{B}||\mathcal{B}^{-b}|\sum_{t'=t-F+1}^{t}(\Delta^{\uparrow}\mathcal{I}_{*,*}(t')\\&-\mathbb{1}[t'|F]\Delta^{\downarrow}\mathcal{I}_{*,*}(t')),\end{aligned}\tag{32}$$

where the convergence speed is decided by local information ($\mathcal{I}_{*,\text{env}}$) and cooperation information ($\mathcal{I}_{*,*}$) separately. In the following, we analyse the convergence speed of the aforementioned two parts individually, and the convergence speed is decided by the larger one between these two results.

First, we analyse the convergence speed of the cooperation information part. By substituting Eq.(26) into Eq.(32), the part of information gain for bth agent at time t is denoted as

$$\begin{aligned}\Delta^{\uparrow}\mathcal{I}_b(t)&=\Delta^{\uparrow}\mathcal{I}_{*,\text{env}}(t)+|\mathcal{B}^{-*}|\Delta^{\uparrow}\mathcal{I}_{*,*}(t)\\&=\mathcal{K}_{\text{env}}\Lambda(\mathcal{C}_{\text{env}}-\mathcal{I}_{*,\text{env}}(t-1))\\&\quad+|\mathcal{B}^{-*}|\mathcal{K}_*\Lambda(\mathcal{C}_*-\mathcal{I}_{*,*}(t-1)).\end{aligned}\tag{33}$$

Similarly, we can expand the cooperation information loss between agent b and b' as Eq.(34) [33].

$$\begin{aligned}\Delta^{\downarrow}\mathcal{I}_{*,*}(t)&=\frac{|\mathcal{B}^{-*}|\mathcal{K}_*\Lambda(\mathcal{C}_*-\mathcal{I}_{*,*}(t-1))}{\mathcal{I}_{b'}(t-1)+|\mathcal{B}^{-*}|\mathcal{K}_*\Lambda(\mathcal{C}_*-\mathcal{I}_{*,*}(t-1))}\\&\quad\mathcal{I}_{*,*}(t-1).\end{aligned}\tag{34}$$

Substituting Eq.(34) and cooperation information term in Eq.(33) into Eq.(32), the overall gain for cooperation information $\mathcal{I}_{*,*}$ between FedAvg operations (including $F-1$ local update and a FedAvg update) can be denoted as Eq.(35).

$$\begin{aligned}&\mathcal{I}_{*,*}(t)-\mathcal{I}_{*,*}(t-F)=\\&|\mathcal{B}|\sum_{t'=t-F+1}^{t}\left(\Delta^{\uparrow}\mathcal{I}_{*,*}(t')-\mathbb{1}[t|F]\Delta^{\downarrow}\mathcal{I}_{*,*}(t')\right)\\&\le|\mathcal{B}|\sum_{t'=t-F+2}^{t}\Big[(\mathcal{K}_*(\mathcal{C}_*-\mathcal{I}_{*,*}(t'-1)))(1-\\&\frac{\mathcal{I}_{*,*}(0)}{\mathcal{C}_{*,\text{env}}/|\mathcal{B}^{-*}|+\mathcal{K}_*\mathcal{C}_*+\mathcal{C}_*})\Big]+|\mathcal{B}|\mathcal{K}_*(\mathcal{C}_*-\mathcal{I}_{*,*}(t-F)).\end{aligned}\tag{35}$$

Then, we denote t_F as the closest time instance with FedAvg operation from t, i.e. $t-t_F<F$. By continuing decomposing F steps iteratively between two FedAvg steps with the Eq.(35) with Eq.(26) and Eq.(34), we can get Eq.(36).

$$\begin{aligned}&\mathcal{I}_{*,*}(t)-\mathcal{I}_{*,*}(t_F)\\&\le|\mathcal{B}|\sum_{t'=t_F+2}^{t}(\alpha\mathcal{C}_*-\alpha\mathcal{I}_{*,*}(t'-1))+|\mathcal{B}|\mathcal{K}_*(\mathcal{C}_*-\mathcal{I}_{*,*}(t_F))\\&=|\mathcal{B}|\alpha(t-t_F-1)\mathcal{C}_*-|\mathcal{B}|\alpha\Big[\sum_{t'=t_F+2}^{t-1}\mathcal{I}_{*,*}(t'-1)\\&\quad+\mathcal{I}_{*,*}(t-2)+(\alpha\mathcal{C}_*-\alpha\mathcal{I}_{*,*}(t-2))\Big]\\&\quad+|\mathcal{B}|\mathcal{K}_*(\mathcal{C}_*-\mathcal{I}_{*,*}(t_F))\\&=|\mathcal{B}|[(1-\mathcal{K}_*)(1-(1-\alpha)^{F-1})+\mathcal{K}_*](\mathcal{C}_*-\mathcal{I}_{*,*}(t_F)).\end{aligned}\tag{36}$$

As the FedAvg round F is small compared to the overall learning rounds, we only look at the time after each FedAvg operation. Then, by further expanding the equation to $t=0$, we have the formulation of $\mathcal{I}_{*,*}(t)$ as Eq.(37).

$$\begin{aligned}\mathcal{I}_{*,*}(t)\le\mathcal{C}_*-(1-|\mathcal{B}|[(1-\mathcal{K}_*)(1-\\(1-\alpha)^{F-1})+\mathcal{K}_*])^{\lfloor t/F\rfloor}(\mathcal{C}_*-\mathcal{I}_{*,*}(0)),\end{aligned}\tag{37}$$

where $\alpha=\mathcal{K}_*(1-\frac{\mathcal{I}_{*,*}(0)}{\mathcal{C}_{*,\text{env}}/|\mathcal{B}^{-*}|+\mathcal{K}_*\mathcal{C}_*+\mathcal{C}_*})$.

We solve t for the upper bound t when $\mathcal{I}_{*,*}(t)\le\mathcal{C}_*(1-\epsilon)$ and conclude the proof of Eq.(28).

Similarly, for local information $\mathcal{I}_{*,\text{env}}$, we have the upper bound of its convergence speed as

$$\begin{aligned}\mathcal{I}_{*,\text{env}}(t)&\le\mathcal{I}_{*,\text{env}}(t-1)+\Delta\mathcal{I}_{*,\text{env}}(t)\\&=(1-|\mathcal{B}|\mathcal{K}_*)\mathcal{I}_{*,\text{env}}(t-1)+|\mathcal{B}|\mathcal{K}_*\mathcal{C}_{\text{env}}\\&\le\mathcal{C}_{\text{env}}-(1-|\mathcal{B}|\mathcal{K}_*)^t(\mathcal{C}_{\text{env}}-\mathcal{I}_{*,\text{env}}(0)).\end{aligned}\tag{38}$$

Thus, for $\mathcal{I}_{*,\text{env}}(t)\le\mathcal{C}_{\text{env}}(1-\epsilon)$, we have

$$t_{\text{env}}=\log_{1-|\mathcal{B}|\mathcal{K}_{\text{env}}}(\frac{\mathcal{C}_{\text{env}}\epsilon}{\mathcal{C}_{\text{env}}-\mathcal{I}_{*,\text{env}}(0)}).\tag{39}$$

REFERENCES

[1] F. Guidolin, L. Badia, and M. Zorzi, "A distributed clustering algorithm for coordinated multipoint in LTE networks," *IEEE Wireless Commun. Lett.*, vol. 3, no. 5, pp. 517–520, Jul. 2014.

[2] S. Bassoy, H. Farooq, M. A. Imran, and A. Imran, "Coordinated multipoint clustering schemes: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 743–764, 2nd Quart., 2017.

[3] S. S. Ali and N. Saxena, "A novel static clustering approach for CoMP," in *Proc. 7th Int. Conf. Comput. Converg. Technol. (ICCCT)*, Dec. 2012, pp. 757–762.

[4] H. Sun, X. Zhang, and W. Fang, "Dynamic cell clustering design for realistic coordinated multipoint downlink transmission," in *Proc. IEEE 22nd Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2011, pp. 1331–1335.

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

[6] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 101–115, Nov. 2021.

[7] H. Shiri, J. Park, and M. Bennis, "Communication-efficient massive UAV online path control: Federated learning meets mean-field game theory," 2020, *arXiv:2003.04451*.

[8] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, Nov. 2021.

[9] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," 2017, *arXiv:1706.02275*.

[10] H. He, J. Boyd-Graber, K. Kwok, and H. Daumé III, "Opponent modeling in deep reinforcement learning," 2016, *arXiv:1609.05559*.

[11] A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1226–1252, 2nd Quart., 2021.

[12] Y. Al-Eryani, M. Akrout, and E. Hossain, "Multiple access in cell-free networks: Outage performance, dynamic clustering, and deep reinforcement learning-based design," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1028–1042, Apr. 2021.

[13] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, "Distributed multi-agent meta learning for trajectory design in wireless drone networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3177–3192, Oct. 2021.
[14] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," 2018, *arXiv:1803.11485*.
[15] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," Feb. 2015, *arXiv:1502.05477*.
[16] F. L. D. Silva and A. H. R. Costa, "A survey on transfer learning for multiagent reinforcement learning systems," *J. Artif. Intell. Res.*, vol. 64, pp. 645–703, Mar. 2019.
[17] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," 2018, *arXiv:1802.05438*.
[18] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248–1261, Jun. 2019.
[19] Y. Wang, B. Han, T. Wang, H. Dong, and C. Zhang, "Off-policy multi-agent decomposed policy gradients," 2020, *arXiv:2007.12322*.
[20] J. D. Co-Reyes, S. Sanjeev, G. Berseth, A. Gupta, and S. Levine, "Ecological reinforcement learning," 2020, *arXiv:2006.12478*.
[21] A. Naik, R. Shariff, N. Yasui, H. Yao, and R. S. Sutton, "Discounted reinforcement learning is not an optimization problem," 2019, *arXiv:1910.02140*.
[22] P. Georgakopoulos, T. Akhtar, I. Politis, C. Tselios, E. Markakis, and S. Kotsopoulos, "Coordination multipoint enabled small cells for coalition-game-based radio resource management," *IEEE Netw.*, vol. 33, no. 4, pp. 63–69, Jul. 2019.
[23] T. M. Shami, D. Grace, A. Burr, and M. D. Zakaria, "User-centric JT-CoMP clustering in a 5G cell-less architecture," in *Proc. 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 177–181.
[24] M. Sohaib J. Solaija, H. Salman, A. B. Kihero, M. I. Saglam, and H. Arslan, "Generalized coordinated multipoint framework for 5G and beyond," 2020, *arXiv:2008.06343*.
[25] K. Zhang, Z. Yang, and T. Başar, "Decentralized multi-agent reinforcement learning with networked agents: Recent advances," 2019, *arXiv:1912.03821*.
[26] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor-critic algorithms," *Automatica*, vol. 45, no. 11, pp. 2471–2482, Nov. 2009.
[27] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin, "Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 177–191, Sep. 2020.
[28] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," 2017, *arXiv:1707.06887*.
[29] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," 2016, *arXiv:1612.01939*.
[30] V. R. Konda and J. N. Tsitsiklis, "Convergence rate of linear two-time-scale stochastic approximation," *Ann. Appl. Probab.*, vol. 14, no. 2, pp. 796–819, 2004.
[31] H. L Prasad, L. A. Prashanth, and S. Bhatnagar, "Actor-critic algorithms for learning Nash equilibria in N-player general-sum games," 2014, *arXiv:1401.2086*.
[32] R. B. Diddigi, S. K. R. Danda, K. J. Prabuchandran, and S. Bhatnagar, "Actor-critic algorithms for constrained multi-agent reinforcement learning," 2019, *arXiv:1905.02907*.
[33] J. K. Terry, N. Grammel, S. Son, and B. Black, "Parameter sharing for heterogeneous agents in multi-agent reinforcement learning," 2020, *arXiv:2005.13625*.
[34] Q. Wu, L. Liu, and S. Xue, "Global update guided federated learning," 2022, *arXiv:2204.03920*.
[35] V. Smith, S. Forte, M. Chenxin, M. Takáč, M. I. Jordan, and M. Jaggi, "CoCoA: A general framework for communication-efficient distributed optimization," *J. Mach. Learn. Res.*, vol. 18, p. 230, Jul. 2018.
[36] M. Hessel et al., "Rainbow: Combining improvements in deep reinforcement learning," Oct. 2017, *arXiv:1710.02298*.
[37] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," Sep. 2015, *arXiv:1509.02971*.
[38] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Autonomous Agents and Multiagent Systems*, G. Sukthankar and J. A. Rodriguez-Aguilar, Eds. Cham, Switzerland: Springer, 2017, pp. 66–83.

Fenghe Hu is currently pursuing the Ph.D. degree with the Center for Telecommunications Research (CTR), King's College London. His research interests include providing external reality service via wireless connection.

Yansha Deng (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Queen Mary University of London, U.K., in 2015. From 2015 to 2017, she was a Post-Doctoral Research Fellow with King's College London, U.K., where she is currently a Senior Lecturer (an Associate Professor) with the Department of Engineering. Her research interests include molecular communication and machine learning for 5G/6G wireless networks. She has served as a TPC Member for many IEEE conferences, such as IEEE GLOBECOM and ICC. She was a recipient of the Best Paper Awards from ICC 2016 and GLOBECOM 2017 as the first author and the IEEE Communications Society Best Young Researcher Award for the Europe, Middle East, and Africa Region 2021. She also received the Exemplary Reviewers of the IEEE TRANSACTIONS ON COMMUNICATIONS in 2016 and 2017 and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS in 2018. She is also an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE TRANSACTIONS ON MOLECULAR, BIOLOGICAL AND MULTI-SCALE COMMUNICATIONS, a Senior Editor of the IEEE COMMUNICATIONS LETTERS, and the Vertical Area Editor of *IEEE Internet of Things Magazine*.

A. Hamid Aghvami (Life Fellow, IEEE) joined King's College London, an Academic Staff, in 1984. In 1989, he was promoted to a Reader, and was promoted to a Professor of telecommunications engineering in 1993. He was a Visiting Professor at NTT Radio Communication Systems Laboratories in 1990, a Senior Research Fellow at BT Laboratories from 1998 to 1999, and was an Executive Advisor of Wireless Facilities Inc., USA, from 1996 to 2002. He was the Director of the Centre from 1994 to 2014. He is currently the Founder of the Centre for Telecommunications Research, King's College London. He is the Chairperson of Advanced Wireless Technology Group Ltd. He is also the Managing Director of Wireless Multimedia Communications Ltd., his own consultancy company. He is also a Visiting Professor at Imperial College London. He carries out consulting work on digital radio communications systems for British and international companies. He has published over 580 technical journals and conference papers, filed 30 patents, and given invited talks and courses the world over on various aspects of mobile radio communications. He leads an active research team working on numerous mobile and personal communications projects for fourth and fifth generation networks; these projects are supported both by government and industry. He was a member of the Board of Governors of the IEEE Communications Society from 2001 to 2003, was a Distinguished Lecturer of the IEEE Communications Society from 2004 to 2007, and has been a member, the Chairperson, and the Vice-Chairperson of the technical program and organizing committees of a large number of international conferences. He is a fellow of the Royal Academy of Engineering and a fellow of the IET. He was awarded the IEEE Technical Committee on Personal Communications (TCPC) Recognition Award in 2005 for his outstanding technical contributions to the communications field, and for his service to the scientific and engineering communities. In 2009, he was awarded a Fellowship of the Wireless World Research Forum in recognition of his personal contributions to the wireless world, and for his research achievements as the Director of the Centre for Telecommunications Research, King's College London. He is also the Founder of the International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), a major yearly conference attracting some 1,000 attendees.