# Meta Federated Reinforcement Learning for Distributed Resource Allocation

Zelin Ji, *Graduate Student Member, IEEE*, Zhijin Qin, *Senior Member, IEEE*, and Xiaoming Tao, *Senior Member, IEEE*

***Abstract*—In cellular networks, resource allocation is usually performed in a centralized way, which brings huge computation complexity to the base station (BS) and high transmission overhead. This paper introduces a distributed resource allocation method that aims to maximize energy efficiency (EE) while ensuring quality of service (QoS) for users. Specifically, to address the challenge of fast-varying wireless channel conditions, we propose a robust meta federated reinforcement learning (*MFRL*) framework that enables local users to optimize transmit power and assign channels using locally trained neural network models. This approach offloads the computational burden from the cloud server to the local users, reducing transmission overhead associated with local channel state information. The BS performs the meta-learning procedure to initialize a general global model, enabling rapid adaptation to different environments and improved EE performance. The federated learning technique, based on decentralized reinforcement learning, promotes collaboration and mutual benefits among users. Analysis and numerical results demonstrate that the proposed *MFRL* framework accelerates the reinforcement learning process, decreases transmission overhead, and offloads computation, while outperforming the conventional decentralized reinforcement learning algorithm in terms of convergence speed and EE performance across various scenarios.**

***Index Terms*—Federated learning, meta-learning, reinforcement learning, resource allocation.**

## I. Introduction

The wireless network industry is experiencing an undeniable trend of development. The third Generation Partnership Project (3GPP) has standardized the access technique and physical channel model for the fifth-generation new radio (5G NR) network. This standard enables user equipment (UE) to dynamically switch between resource blocks (RBs) with varying bandwidths and supports multiple subcarrier spacing [2], [3]. Building upon the foundation established by 5G, the next generation of networks, such as sixth generation (6G) and beyond, aim to provide enhanced and augmented services of 5G NR while transitioning towards decentralized, fully autonomous, and remarkably flexible user-centric systems [4]. These emerging techniques impose more stringent requirements on decentralized resource allocation methods, emphasizing the significance of optimizing RB assignments to enhance the overall quality of service (QoS) within the systems.

Nevertheless, the fast variations and rapid fluctuations in channel conditions render conventional resource allocation approaches reliant on perfect channel state information (CSI) impractical [5]. The inherent non-convexity of the resource allocation problem resulting from discrete resource block association necessitates computationally demanding solutions. Furthermore, the coupled variables further exacerbate the complexity of the problem. Traditionally, resource allocation problems have been addressed through matching algorithms executed at the central base station (BS), resulting in substantial computational burdens on the cloud server. All of the aforementioned challenges require a brand-new optimization tool capable of effectively operating in unstable wireless environments.

Machine learning (ML) methods, particularly deep learning (DL) approaches, have emerged as promising tools to tackle mathematically intractable and high-computational problems. However, artificial neural networks (NNs) typically demand massive amounts of training data, even for simple binary classification tasks. Additionally, the issue of overfitting makes artificial NNs difficult to adapt and generalize when encountering new environments, necessitating additional data for model retraining and affecting the efficiency of training data. In particular, the fast channel variations and flexible network structure in 5G beyond network services restrict the application of conventional ML algorithms.

To enable fast and flexible learning, meta learning has been proposed. Meta learning allows the model to adapt to new tasks with faster convergence speed by leveraging input from experience gained from different training tasks [6]–[8]. For example, model-agnostic meta-learning (MAML) [8] is a meta-learning technique that integrates prior experience and knowledge from the new environment, empowering models with the ability to generalize and rapidly adapt to new tasks. Another approach to improve data efficiency is through experience sharing among models, known as federated learning. With periodic local model averaging at the cloud BS, federated learning enables local users to collectively train a global model using their raw data while keeping the data locally stored on the mobile devices [9]. This paper focuses on meta-learning enabled federated reinforcement learning, with the objective of improving the performance of the reinforcement learning algorithm for resource allocation tasks in wireless communications.

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62293484 and 61925105. Part of this work was presented at the IEEE International Conference on Communications 2022 [1]. (*Corresponding author: Zhijin Qin*.)

Zelin Ji is with School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (email: z.ji@qmul.ac.uk).

Zhijin Qin and Xiaoming Tao are with Department of Electronic Engineering, Tsinghua University, Beijing, China. (email: qinzhijin@tsinghua.edu.cn; taoxm@tsinghua.edu.cn).

Through the implementation of periodic local model averaging at the cloud-based base station (BS), federated learning facilitates collaborative training of a global model by enabling local users to utilize their respective raw data, which remains stored locally on their mobile devices [9]. This paper investigates the application of meta-learning within the context of federated reinforcement learning, with the aim of enhancing the performance of the reinforcement learning algorithm in resource allocation tasks within wireless communication systems.

### A. Related work

*1) Energy-Efficient Resource Allocation:* Presently, most cellular user equipment (UE) operates on battery power, and the use of rate maximization-oriented algorithms [10] may result in unnecessary energy consumption, which is unfavorable for the advancement of massive capacity and connectivity in 5G and beyond communications.

Existing literature on energy-efficient resource allocation primarily focuses on optimizing transmit power and channel assignment [11]–[13]. Robat Mili *et al.* [11] concentrate on maximizing energy efficiency (EE) for device-to-device communications. While numerous studies have investigated resource allocation in wireless communication systems, most of them rely on centralized approaches, which are complex and not easily scalable [13]. In such centralized approaches, the central entity needs global channel state information (CSI) to assign channels to UEs, leading to significant communication overhead and latency. Consequently, distributed low-complexity algorithms are preferable over centralized ones.

Game theory has been adopted for decentralized resource allocation [13]–[15]. However, these approaches typically assume a static radio environment and require multiple iterations for UEs to converge to the Nash Equilibrium (NE) point. In the practical environment, the performance of game theory based algorithms is impacted by the rapid fluctuations in the wireless channel. Yang *et al.* [14] and Dominic *et al.* [15] integrate the game theory and stochastic learning algorithm (SLA) to enable local users to learn from past experience and adapt to channel variations. Yan *et al.* [16] further investigate resource allocation for semantic communications, where the channel assignment and power allocation problem is modeled as a matching game among users. Nevertheless, game theory based algorithms do not fully explore the advantages of collaboration and communication among users, potentially affecting system-level performance.

*2) Decentralized Reinforcement Algorithms in Wireless Communications:* A promising solution to address concerns regarding complexity and signaling cost concerns involves establishing a decentralized framework for resource allocation and extending the intelligent algorithms to encompass cooperative large-scale networks. The adoption of multi-agent reinforcement learning (MARL) algorithm presents an opportunity to tackle the challenges associated with complexity and enhance the intelligence of local UEs. MARL algorithms rely solely on real-time local information and observations, thereby significantly reducing communication overhead and latency. Mathematically, MARL can be formulated as a Markov decision process (MDP), where training agents observe the current state of the environment at each step and determine an action based on the current policy. Agents receive corresponding rewards that evaluate the immediate impact of the chosen state-action pair. The policy updates are based on the received rewards and the specific state-action pair, and the environment transitions to a new state subsequently. The application of MARL approaches in wireless communications has been extensive [17]–[20]. Wang *et al.* [18] have demonstrated that such a decentralized optimization approach can achieve near-optimal performance. Ji *et al.* [20] further have extended the application of the MARL algorithm for the joint optimization of the communication and computation resources for semantic communications. However, local user equipment (UE) cannot directly access global environmental states, and UEs are unaware of the policies adopted by other UEs. Consequently, there is a possibility that UEs may select channels already occupied by other UEs, leading to transmission failures in the orthogonal frequency-division multiple access (OFDMA) based schemes.

*3) Reinforcement Algorithm for Jointly Resource Optimization:* It is noted that the resource block association problem is a discrete optimization problem, which is usually solved by value-based methods, e.g., Q-learning, SARSA, and Deep Q-learning. Meanwhile, the transmit power is the continuous variable, and only policy-based algorithm can deal with the continuous optimization. Hence, how to jointly optimize the transmit power and channel assignment becomes a challenge. In some work, the transmit power is approximated to discrete power levels, and the user can only transmit by these presetting power levels [1], [21]. However, discrete transmit power with large intervals means performance reduction. On the other hand, the complexity could be very high if the number of power levels is significant. To address these concerns, Yuan *et al.* [22] proposed a framework with a combination of value-based network and policy-based network. Similarly, Hehe *et al.* [23] also proposed a combination framework with different components to address the discrete user association problem and continuous power allocation problem. However, in such works the different networks are trained simultaneously, which leads to an unstable framework and makes the NNs hard to train and converge.

### B. Motivations and Contributions

*1) Federated Reinforcement Learning:* The primary obstacle faced by MARL algorithms is the instability and unpredictability of actions taken by other user equipment (UEs), resulting in an unstable environment that affects the convergence performance of MARL [24]. Consequently, a partially collaborative MARL structure with communication among UEs becomes necessary. In this structure, each agent can share its reward, RL model parameters, action, and state with other agents. Various collaborative RL algorithms may employ different information-sharing strategies. For instance, some collaborative MARL algorithms require agents to share their state and action information, while others necessitate the

sharing of rewards. The training complexity and performance of a collaborative MARL algorithm are influenced by the data size that each agent needs to share. This issue becomes severer when combining neural networks (NN) with reinforcement learning. In a traditional centralized reinforcement algorithm, e.g., deep Q-network (DQN), the environment's interactive experiences and transitions are stored in the replay memory and utilized to train the DQN model. However, in multi-agent DQN, local observations fail to represent the global environment state, significantly diminishing the effectiveness of the replay memory. Although some solutions have been proposed to enable replay memory for MARL, these approaches lack scalability and fail to strike a suitable balance between signaling costs and performance.

To address the issue of non-stationarity, it is necessary to ensure the sharing of essential information among UEs, which can be facilitated by federated learning [25]. Federated learning has demonstrated successful applications in tasks such as next-word prediction [26] and system-level design [27]. Specifically, federated reinforcement learning (FRL) enables UEs to individually explore the environment while collectively training a global model to benefit from each other's experiences. In comparison to MARL approaches, the FRL method enables UEs to exchange their experiences, thereby enhancing convergence performance [28]. This concept has inspired the work of Zhang *et al.* [29] in improving WiFi multiple access performance and Zhong *et al.* [30] in optimizing the placement of reconfigurable intelligent surfaces through the application of FRL.

*2) Meta Reinforcement Technique for Fast Adaptation and Robustness:* Another main challenge of the reinforcement learning algorithm is the demand for massive amounts of training data. Since the training data can only be acquired by interacting with the environment, the agent usually needs a long-term learning process until it can learn from a good policy. Moreover, using such a large amount of data to train an agent also may lead to overfitting and restrict the scalability of the trained model. In the scope of the wireless environment, the fast fading channels and unstable user distributions also put forward higher requirements on robustness and generalization ability. Particularly, the previous resource allocation algorithms usually set a fixed number of users, which makes the algorithm lack scalability to various wireless environments in practical implementation.

Meta-learning is designed to optimize the model parameters using less training data, such that a few gradient steps will produce a rapid adaptation performance on new tasks. During the meta-learning training process, the model takes a little training data from different training tasks to initialize a general model, which reduces the model training steps significantly. The meta-learning can be implemented in different ways. Wang *et al.* [6] and Duan *et al.* [7] have applied recurrent NN and the long short-term memory to integrate the previous experience into a hidden layer, and NNs have been adopted to learn the previous policy. Finn *et al.* [8] have leveraged the previous trajectories to update the NNs, and further extended the meta-learning to reinforcement learning. In this paper, we consider the meta-learning for initializing the NNs for MARL. In the scope of wireless communications, Yuan *et al.* [22] have adopted the meta reinforcement learning for different user distributions and confirm that the meta reinforcement learning is a better initialization approach and can achieve better performance in new wireless environments.

Another challenge caused by federated learning is the heterogeneity in systems and the non-identical data distributions in RL may slow down or even diverge the convergence of the local model. Inspired by the meta-learning, Fallah *et al.* [31] have developed a combined model, in which the global training stage of the federated learning can be considered as the initialization of the model for meta-learning, and the personalized federated learning stage can be seen as the adaptation stage for meta-learning. Due to the similar mathematical expression, we can combine federated learning and meta-learning naturally, so that training and adapting the models from statistically heterogeneous local RL replay memories. The aforementioned studies serve as valuable inspiration for us to explore the application of meta-learning and FRL in addressing the challenges of channel assignment and power optimization. By leveraging these techniques, we aim to distribute the computational load to local user equipment (UEs), reduce transmission overhead, and foster collaboration among UEs.

This paper introduces a novel framework that combines meta-learning and FRL for distributed solutions to the channel assignment and power optimization problem. To the best of our knowledge, this is the first endeavor to integrate meta-learning and FRL in the context of resource allocation in wireless communications. The contributions of this paper are summarized as follows:

1) A meta federated reinforcement learning framework, named *MFRL*, is proposed to jointly optimize the channel assignment and transmit power. The optimization is performed distributed at local UEs to lower the computational cost at the BS and the transmission overhead.
2) To improve the robustness of the proposed algorithm, we leverage the meta-learning to initialize a general model, which can achieve fast adaptation to new resource allocation tasks and guarantee the robustness of the proposed *MFRL* framework.
3) To address the joint optimization of the discrete and continuous variables, we redesign the action space for the RL algorithm and design the corresponding proximal policy optimization (PPO) network to optimize the real-time resource allocation for each UE.
4) To explore the collaboration among cellular users, we propose a global reward regarding the sum EE and the successful allocation times for all UEs and apply the *MFRL* framework for enabling experience sharing among UEs.

The remainder of the paper is organized as follows. In Section II, the system model is presented and an EE maximization problem is formulated. The proposed *MFRL* framework is presented in Section III. The numerical results are illustrated in Section IV. The conclusion is drawn in Section V.

## II. System Model

In this paper, we assume that the set of UEs is denoted as $\mathcal{UE} = \{UE_1, \ldots, UE_I\}$, where $I$ is the total number of UEs. For $UE_i$, the binary channel assignment vector is given by $\boldsymbol{\rho}_i = [\rho_{i,1}, \ldots, \rho_{i,n}, \ldots, \rho_{i,N}], i \in I, n \in N$, where $N$ is the number of subchannels. The channel assignment parameter $\rho_{i,n} = 1$ indicates that the $n$-th subchannel is allocated to $UE_i$, otherwise $\rho_{i,n} = 0$. Each UE can only accesses one channel, i.e., $\sum_{n=1}^{N} \rho_{i,n} = 1, \forall i \in I$. Meanwhile, we consider a system with OFDMA, which means a channel can be accessed by at most one UE within a cluster, i.e., $\sum_{i=1}^{I} \rho_{i,n} \in \{0,1\}, \forall n \in N$. In the case of each user equipment (UE), successful transmission with the base station (BS) is achieved when the UE accesses a specific subchannel without any other UEs within the same cluster accessing the same subchannel. For the cases where the number of UEs is larger than the channels, the non-orthogonal multiple access technique needs to be applied to support multiple UEs to access the same channel, which is beyond the scope of this article. To ensure that the channel assignment problem can be solved and simplify the system settings, we assume that at most $N$ UEs can access the channel and transmit the data within a time slot, i.e., $N >= I$. Consequently, if each UE is allocated a channel that does not conflict with other UEs within the cluster, this allocation is considered a successful channel assignment.

The pathloss of a common urban scenario with no line of sight link between $UE_i$ and the BS can be denoted by [3]

$$PL_{i,n} = 32.4 + 20\log_{10}(f_n) + 30\log_{10}(d_{i,n})\,(\mathrm{dB}), \quad (1)$$

where $d_{i,n}$ represents the 3D distance between $UE_i$ and the BS, $f_n$ represents the carrier frequency for $n$-th subchannel. Considering the small-scale fading, the overall channel gain can be thereby denoted by

$$h_{i,n} = \frac{1}{10^{(PL_{i,n}/10)}} \psi m_n, \quad (2)$$

where $\psi$ is the log-normally distributed shadowing parameter. According to the aforementioned pathloss model, there is no line of sight between UEs and the BS, and $m_n$ represents the Rayleigh fading power component of the $n$-th subchannel. Hence, the corresponding signal-to-noise ratio (SNR) between the BS and $UE_i$ transmitting over the $n$-th subchannel is represented as

$$\gamma_{i,n} = \frac{\rho_{i,n} h_{i,n} p_i}{N_n}, \quad (3)$$

where $N_n = W_n \sigma_n^2$ represents the Gaussian noise power on the $n$-th subchannel. The uplink EE for a successful channel assignment of $UE_i$ is given by

$$u_{i,n} = \begin{cases} \frac{W_n}{p_i + p_i^c} \log_2(1 + \gamma_{i,n}), & \text{if } \sum_{\mathrm{n}=1}^{\mathrm{N}} \rho_{\mathrm{i,n}} = 1; \\ 0, & \text{else.} \end{cases} \quad (4)$$

where $W_n = k_n \times b_n$ is the bandwidth of the $n$-th subchannel, $k_n$ represents the number of subcarriers in each subchannel, and $b_n$ denotes the subcarriers spacing for $n$-th subchannel. The static and circuit power for maintaining the basic operations of the UE system is set to a constant value $p_i^c$. Meanwhile, for the unsuccessful assignment, i.e., the UE cannot access any subchannel, the uplink rate is set to 0 as it is unacceptable for the OFDMA system.

The EE maximum problem is formulated as

$$\begin{aligned} (\mathbf{P0}) \ \underset{\{\boldsymbol{\rho}, \boldsymbol{p}\}}{\text{maximize}} \quad & \sum_{i=0}^{I} \sum_{n=0}^{N} u_{i,n} && (5a) \\ \text{subject to} \quad & p_i \le p_{max}, \forall i \in I, && (5b) \\ & \gamma_{i,n} > \gamma_{min}, \forall i \in I, && (5c) \\ & \sum_{n=1}^{N} \rho_{i,n} = 1, \forall i \in I, && (5d) \\ & \sum\nolimits_{i=1}^{I} \rho_{i,n} \in \{0,1\}, \forall n \in N. && (5e) \end{aligned}$$

where $\boldsymbol{p} = \{p_1, \ldots, p_I\}$ denotes the transmit power vector of UEs, $\gamma_{min}$ represents the minimum SNR requirement to guarantee the QoS for UEs. Constraint (5d) and (5e) make the EE maximization problem a non-convex optimization problem and cannot be solved by mathematical convex optimization tools. In the literature, channel allocation problems are usually formed as linear sum assignment programming (LSAP) problems. To solve this problem, local CSI or the UE related information, e.g., location and velocity should be uploaded to the BS, then the centralized Hungarian algorithm [32] can be invoked to solve the problem with computational complexity $O\left(I^3\right)$. The computational complexity grows exponentially with the number of UEs, and the mobility of UEs causes the variable CSI, which means the high-complexity algorithm needs to be executed frequently, leading to high transmission overhead and high computational pressure to the BS. Moreover, due to the transmission latency, the current optimized resource allocation scheme by the BS may not be optimal for UEs anymore, and a distributed and low complexity resource allocation approach on the UE side is more than desired.

According to the constraint (5d) and (5e), each UE can only access one subchannel, and it is clear that the subchannel assignment is a discrete optimization problem. As aforementioned concerns in Section I, it is hard to train different types of neural networks simultaneously. In another way, the discrete assignment problem can be described by different probabilities to choose different subchannels, and then one-dimensional discrete choice can be mapped to high-dimensional probability distributions. Overall, the joint optimization problem can be solved by a simple policy-based framework with a specific output design.

## III. Proposed Meta Federated Reinforcement Learning for Resource Allocation

In this section, we will first introduce the proposed *MFRL* framework from an overall perspective. Then we will design the NN structure to solve this EE maximization problem, and propose a meta reinforcement learning scheme for the NN initialization. We also demonstrate the meta-training and meta-adapting algorithms in detail. Finally, we will present the federated learning algorithm and procedures.

The proposed algorithm starts from the meta-training for initializing the global generalized model at the BS. The initial

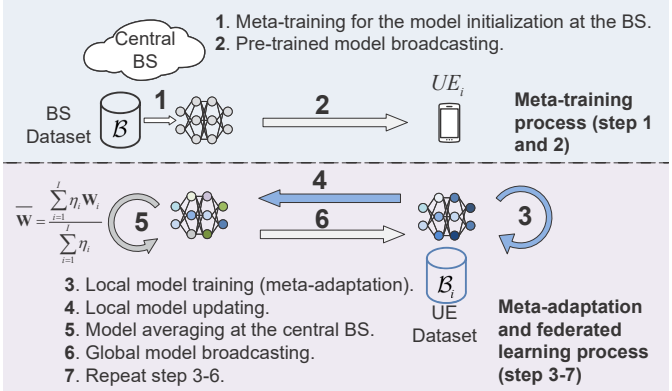



Fig. 1. The proposed *MFRL* framework. The local models are uploaded and averaged periodically.

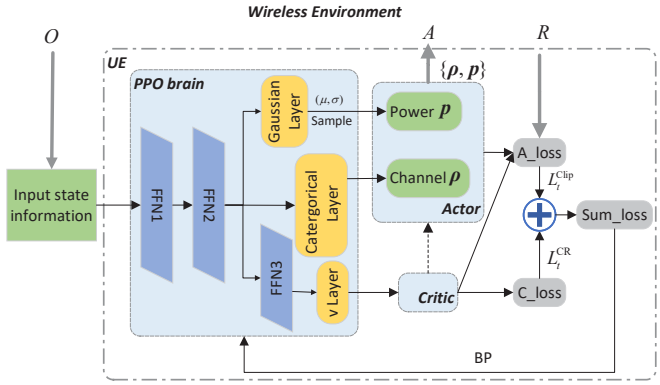



Fig. 2. The proposed PPO network structure for the *MFRL* framework.

model is meta-trained using the BS data set. After the initial global model is trained, it will be broadcast to the local UEs for adapting to the new environments. During the meta-adapting, i.e., the fine-tuning process, the local models are trained using a local database, i.e., local CSI, and the local models can be reunited as a global model so that the UEs could learn the knowledge from the experiences of other UEs and improve the global EE. One popular way is to average the distributed models and form a global model, which is called federated learning [25]. After the local models are averaged by the BS, it would be broadcast to the local UEs which will fine-tune the global model and adapt to the local scenarios. This process will be repeated until the meta-adaptation stage finishes. The overall procedure is shown in Fig. 1

### A. Neural Network Structure Design

As the aforementioned description, the resource allocation problem can be modeled as a multi-agent markov decision process (MDP), which is mathematically expressed by a tuple, $\langle I, \mathcal{O}, \mathcal{A}, \mathcal{R}, P\rangle$, where $I$ is the number of agents, $I = 1$ degenerates to a single-agent MDP, $\mathcal{O}$ is the combination set of all observation state, $\mathcal{A} = \mathcal{A}_0 \times \cdots \times \mathcal{A}_I$ is the set of actions for each agent, $\mathcal{R}$ is the reward function, which is related to current observation $O_t = \{o_0, \ldots, o_I\} \in \mathcal{O}$, $A_t = \{a_0, \ldots, a_I\} \in \mathcal{A}$, and $O_{t+1} \in \mathcal{O}$. Transition probability function is defined as $P : \mathcal{O} \times \mathcal{A} \to \mathcal{P}(\mathcal{O})$, with $P(O_{t+1}|O_t, A_t)$ being the probability of transitioning into state $O_{t+1}$ if the environment start in state $O_t$ and take joint action $A_t$.

One of the challenges of using deep reinforcement learning algorithms to solve the problem (**P0**) is that the resource allocation of the transmit power and subchannel association is the hybrid optimization of the continuous and discrete variables. As the analysis above, the discrete subchannel association parameter can be described by different probabilities to choose different subchannels, thus the discrete variable can be expressed by probability distributions on subchannels, which is generated by a categorical layer. Meanwhile, continuous power optimization is performed by the Gaussian layer, where the mean and variance of the transmit power can be trained.

In fact, any deep reinforcement learning algorithms with continuous action space can be applied for training the proposed network structure. Specifically, we apply the PPO algorithm because of its ease of use and robustness, which make it the default algorithm by OpenAI [33]. It is noted that the NN architecture shares parameters between the policy and value function, so that the actor network and critic network share the underlying features in the NN, and simplify the meta-learning initialization and model broadcast costs. The corresponding network structure of the local models is illustrated in Fig. 2.

In this paper, we define the observation state at training step $t$ for the UEs, which are considered as the agents in the *MFRL* framework, as $o_{t,i} = \{\{h_{i,n}\}_{\forall n \in N}, t\}$ with dimension $|o_i|$, where $t$ represents the number of epoch. The variables $t$ can be treated as a low-dimensional *fingerprint* information to contain the policy of other agents [24], thus enhancing the stationary and the convergence performance of the *MFRL* algorithm.

The action $a_{t,i}$ for the $UE_i$ including the subchannel and the transmit power choice with dimension $|a| = 2$. The Actor network contains a categorical layer with $N$ neurons to determine which subchannel the local UE should access. The continuous transmit power is optimized by a $\boldsymbol{\sigma}$ layer and a $\boldsymbol{\mu}$ layer, and the power is sampled according to the probability distribution $N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$.

Since we aim to maximize the sum EE of the cellular network, here we design a global reward $r_t$, according to the joint action $\boldsymbol{a}_t$ such that encouraging collaboration of UEs. The global reward at training step $t$ can be defined as

$$r_t = \begin{cases} \sum\limits_{i=0}^{I} r_i(t) & \text{if } \sum\limits_{\mathrm{i}=0}^{\mathrm{I}} \rho_{\mathrm{i,n}} \in \{0,1\}, \forall \mathrm{i} \in \mathrm{I}, \forall \mathrm{n} \in \mathrm{N}; \\ \frac{I^{suc}-I}{I}, & \text{Otherwise}, \end{cases} \tag{6}$$

where $I^{suc}$ denotes the number of UEs that satisfy the subchannel assignment constraints, i.e., $\sum_{i=0}^{I} \rho_{i,n} \in \{0,1\}, \forall n \in N$. For the assignment that fails to meet the subchannel access requirements, a punishment is set to proportional to the number of failure UEs. [1] Meanwhile, the reward for a

[1]Please note that the reward is designed as a sum of EE and the punishment, which makes it a dimensionless parameter and we only need to focus on its value.

successful subchannel assignment is expressed by

$$r_i(t) = \begin{cases} \xi u_{i,n}(t), & \text{if} \gamma_{\text{i,n}} \geq \gamma_{\min}; \\ \xi u_{i,n}^{p_{max}}(t), & \text{if} \gamma_{\text{i,n}}^{\text{Pmax}} \geq \gamma_{\min} > \gamma_{\text{i,n}}; \\ 0, & \text{Otherwise}, \end{cases} \tag{7}$$

where $\xi$ is a constant coefficient, $u_{i,n}^{p_{max}}(t)$ denotes the EE by the maximum transmit power, which means if the UE fails to meet the SNR constraint, it needs to use the maximum transmit power to avoid transmission failure. The UE could receive a reward that is proportional to the achieved EE when satisfying the SNR constraint. The success rate of $UE_i$ can be defined as $\eta_i = \beta_i/T$, where $\beta_i$ represents the successful resource assignment counts for $UE_i$, and $T$ represents the number of resource allocation counts since the initialization the system.

The objective of the proposed *MFRL* framework is to enable UEs to learn a strategy that maximizes the discount reward, which can be expressed by

$$R(\tau) = \sum_{t=0}^{\infty} \xi^t r_t, \tag{8}$$

where $\tau = (o_0, a_0, ..., o_{T+1})$ is a trajectory, $T$ is the current timestamp, $\xi \in (0, 1)$ represents the discount rate, which denotes the impact of the future reward to the current action.

### *B. Policy Gradient in Meta-training*

In the previous work [1], [17], [18], the number of UEs in each cluster is fixed, and the training and testing performance are implemented in the same environment. Particularly, the local model is trained by each UE individually for the *MFRL* algorithm, which limits its application, making it hard to adapt to more complicated practical scenarios. The resource allocation model should have the ability to adapt and generalize to different wireless communication environments with different cluster sizes. Hence, the meta reinforcement learning algorithm can be considered to meet the requirement of the generalization.

The meta-learning can be implemented in different ways, and we apply the MAML method for reinforcement learning [8]. As an instance of the MAML algorithm, the proposed *MFRL* algorithm combines the MAML algorithm with the PPO reinforcement learning and the federated learning algorithms, enhancing its adaptation in the considered wireless communication scenarios. Particularly, the MAML algorithm for reinforcement learning algorithm is divided into two stages, the meta-training stage and the meta-adapting stage, which are detailed in **Algorithm 1** and **Algorithm 2**, respectively. The meta-training stage takes the experience from different tasks, i.e., the resource allocation for different cluster sizes, to initialize a model that can be adopted by UEs in different scenarios and achieve fast adaptation. In the meta-adaptation stage, the local UEs adapt the initialized model for different tasks based on the interaction with various scenarios.

To take the number of UEs into account, the local observation should include the total number of UEs, i.e., $o_{t,i} = \{\{h_{i,n}\}_{\forall n \in N}, I, t\}$. The task set of resource allocation for UEs is defined as $\mathcal{T} = \{\mathcal{T}^{I_k}\}, \forall k \in K$, where $K$ is the number of tasks, $I_k$ is the number of UEs for task $k$. The meta-training process is implemented at the BS, which can use the previous resource allocation experience for different amount of UEs to meta-train an initial model.

At the end of each training epoch, the BS stores the transitions $e_{t,i}^k = \{(o_{t,i}^k, a_{t,i}^k, r_t^k, o_{t+1,i}^k) | i = 0, 1, \ldots, I_k - 1\}$ acquired from $\mathcal{T}^{I_k}$ in the central dataset. The transitions $e_{t,i} = (o_{t,i}, a_{t,i}, r_t, o_{t+1,i})$ are sampled from $\mathcal{B}$ for calculating the advantage function and the estimated state value function, which are introduced in the following paragraphs. The objective function for training the reinforcement model is to maximize the expected reward for each trajectory as

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta(\tau)}[R(\tau)] = \int_\tau P(\tau|\pi_\theta) R(\tau), \tag{9}$$

where $\pi_\theta$ is the parameterized policy, $P(\tau|\pi_\theta) = P(o_0) \prod_{t=0}^{T-1} P(o_{t+1,i}|o_{t,i}, a_{t,i}) \pi_\theta(a_{t,i}|o_{t,i})$ represents the probability of the trajectory $\tau$, $P(o_{t+1,i}|o_{t,i}, a_{t,i})$ is the state transformation probability, $\pi_\theta(a_{t,i}|o_{t,i})$ is the action choice probability, and $P(o_0)$ is the probability of the initial state $o_0$. To optimize the policy, the policy gradient needs to be calculated, i.e., $\theta_{j+1} = \theta_j + \alpha \nabla_\theta J(\pi_\theta)|_{\theta_j}$, where $\alpha$ is the learning rate or the learning step.

The gradient of the policy can be expressed by a general form as

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[ \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_{t,i}|o_{t,i}) \Phi_{t,i} \right], \tag{10}$$

where $\Phi_{t,i}$ could be denoted as the action-value function $Q^{\pi_\theta}(o, a) = \mathbb{E}_{\tau \sim \pi_\theta(\tau)}[R(\tau)|o_0 = o, a_0 = a]$, which is the expectation reward for taking action $a$ at state $o$. Although we can use the action-value function to evaluate the action is good or bad, the action-value function $Q^{\pi_\theta}(o, a)$ relies on the state and the action, which means an optimal policy under a bad state may have less action-value than an arbitrary action under a better state. To address this issue, we need to eliminate the influence caused by the state. First, we prove that the state influence elimination will not affect the value of the policy gradient [34].

**Lemma 1** (Expected Grad-Log-Prob Lemma)**.** *Given $P^{\pi_\theta}$ is a parameterized probability distribution over a random variable $o$, then $\mathbb{E}_{o \sim P^{\pi_\theta}}[\nabla_\theta \log P^{\pi_\theta}(o)] = 0$.*

*Proof.* For all probability distributions, we have

$$\int_o P^{\pi_\theta}(o) = 1. \tag{11}$$

Take the gradient of both side

$$\nabla_\theta \int_o P^{\pi_\theta}(o) = \nabla_\theta 1 = 0. \tag{12}$$

Thus

$$\begin{aligned}&\mathbb{E}_{o\sim P^{\pi_\theta}}\left[\nabla_\theta \log P^{\pi_\theta}(o)\right]\\&=\int_o P^{\pi_\theta}(o)\nabla_\theta \log P^{\pi_\theta}(o)\\&=\int_o \nabla_\theta P^{\pi_\theta}(o)\\&=\nabla_\theta \int_o P^{\pi_\theta}(o)\\&=0.\end{aligned}$$

□

According to Lemma 1, we can derive that for any function $b(o_t)$ that only depends on the state, $\mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|o)b(o)\right]=0$. Hence, it would cause the same expected value of the policy gradient $\nabla_\theta J(\pi_\theta)$ if we substitute the $b(o)$ into the action-value function $Q^{\pi_\theta}(o,a)$. In fact, we can use the state-value function $V^{\pi_\theta}(o)$ which represents whether the state is good for a higher reward or not. Instead of comparing the action-value function $Q^{\pi_\theta}(o,a)$ of the action $a$ directly, it is more reasonable to substitute the influence of the state into the action-value function. We define the substitution $A^{\pi_\theta}(o,a)=Q^{\pi_\theta}(o,a)-V^{\pi_\theta}(o)$ as the advantage function, which represents whether an action good or bad compared with other actions relative to the current policy. Hence, the value function $\Phi_{t,i}$ can be also denoted as

$$\Phi_{t,i}=Q^{\pi_\theta}(o_{t,i},a_{t,i})-V^{\pi_\theta}(o_{t,i})=A^{\pi_\theta}(o_{t,i},a_{t,i}). \quad (13)$$

### C. Advantage Estimation and Loss Function Design

Although we express the policy gradient by introducing the advantage function, the challenge is, the action-value function and the state-value function cannot be acquired directly from the experience $e_{t,i}$. Instead, the action-value function can be expressed by the temporal difference form [35] as $Q^{\pi_\theta}(o_{t,i},a_{t,i})=r_t+\xi V^{\pi_\theta}(o_{t+1,i})$. In deep reinforcement learning approaches, NNs can be used to estimate the state-value function as $\hat{V}^{\pi_\theta}$, then the estimated advantage function $\hat{A}^{\pi_\theta}(o_{t,i},a_{t,i})=\delta^V_{t,i}=r_t+\xi\hat{V}^{\pi_\theta}(o_{t+1,i})-\hat{V}^{\pi_\theta}(o_{t,i})$ can be derived. However, the bias for this estimation is high, which restricts the training and convergence performance. To overcome this issue, generalized advantage estimation (GAE) [34] can be applied to estimate the advantage function for multi-steps and strike a tradeoff between the bias and variance. The GAE advantage function is denoted by

$$A^{\mathrm{GAE}}(o_{t,i},a_{t,i})=\sum_{l=0}^{T-t}(\lambda\xi)^l\delta^V_{t+l,i}, \quad (14)$$

where $\lambda\in(0,1]$ is the discount factor for reducing the variance of the future advantage estimation.

The actor network is optimized by maximising $L_{AC}=\mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[ratio_{t,i}\times A^{\mathrm{GAE}}(o_{t,i},a_{t,i})\right]$, where $ratio_{t,i}=\frac{\pi_\theta(a_{t,i}|o_{t,i})}{\pi_{\theta_{\mathrm{old}}}(a_{t,i}|o_{t,i})}$ is the action step. However, the large action step could lead to an excessively large policy update, hence we can clip this step and restrict it. The clipped actor objective function is expressed by

$$L_t^{\mathrm{Clip}}=\min\left(ratio_{t,i}\times A^{\mathrm{GAE}}(o_{t,i},a_{t,i}),\ g(\epsilon,A^{\mathrm{GAE}}(o_{t,i},a_{t,i}))\right), \quad (15)$$

where

$$g(\epsilon,A)=\begin{cases}(1+\epsilon)A, & A\geq 0;\\(1-\epsilon)A & A<0,\end{cases} \quad (16)$$

in which the $\epsilon$ is a constant value representing the clip range. The clip operation have been proved to improve the robustness [33].

The loss $L_{\mathrm{CR}}$ for the critic network is to minimize the gap between the estimated state-value function and discount sum reward, which can be expressed by

$$L_t^{\mathrm{CR}}=\left\|r_t+\hat{V}^{\pi_\theta}(o_{t+1,i})-\hat{V}^{\pi_\theta}(o_{t,i})\right\|^2. \quad (17)$$

Combining the objective of the actor network and critic network, we can express the overall objective as

$$L=\arg\min_\theta \mathbb{E}_t\left[L_t^{\mathrm{Clip}}-c_1L_t^{\mathrm{CR}}+c_2E_t\right], \quad (18)$$

where $E_t$ represents an entropy bonus to ensure sufficient exploration, $\theta$ is the weights for the PPO network, $c_1$ and $c_2$ are weight parameters for the estimation of value function and entropy, respectively. Then the initial model will be updated by the stochastic gradient (SG) ascent approach. The details of the meta-training algorithm is shown in **Algorithm 1**.

**Algorithm 1** Meta-training algorithm.

1: **Input**: The task set $\mathcal{T}=\{\mathcal{T}^{I_k}\},\forall k\in K$, BS memory $\mathcal{M}$, BS batch $\mathcal{B}$;
2: Initialize the PPO network $\theta$;
3: **for** each epoch $t$ **do**
4:  **for** each meta task $k$ **do**
5:  The BS acquire the experience $e^k_{t,i}=\{(o^k_{t,i},a^k_{t,i},r^k_t,o^k_{t+1,i})|i=0,1,\ldots,I_k-1\}$ from all UEs and store the transitions in central dataset $\mathcal{M}$;
6:  **end for**
7:  Sample the transitions in the BS batch $\mathcal{B}$;
8:  Update the global PPO network by SG ascent with Adam: $\theta\leftarrow\theta+\alpha_{\mathrm{meta}}\nabla_\theta L$;
9: **end for**
10: **Return**: Pre-trained global model $\theta$.

### D. Meta-Adapting Process

Unlike the meta-training process where the BS stores the transitions and uses these experiences to train a global model, the local UE can train its own model based on its own observations and experience during the meta-adaptation process. Compared with supervised learning which requires sufficient data set and pre-knowledge of the system, the proposed *MFRL* framework can train the local model with the local CSI data which is required by interacting with the environment, thus not only offloading the computational pressure to the UEs, but also lower the transmission overhead significantly.

**Algorithm 2** Meta-adapting algorithm.

**Input**: The pre-trained global model $\theta$, number of UEs $I$, local memory $\mathcal{M}_i$ and batch $\mathcal{B}_i$ for each UE;
Initialize the local models $\theta_{0,i} \leftarrow \theta, \forall i \in I$;
**for** each epoch $j$ **do**
  **for** each D2D pair $i$ **do**
    Collect set of trajectories $\mathcal{M}_i$ by running policy $\pi_{j,i} = \pi(\theta_{j,i})$ in the environment;
    Compute advantage estimations $A^{\mathrm{GAE}}(o_{j,i}, a_{j,i})$ based on current state-value function $\hat{V}^{\pi_\theta}(o)$ and reward $r_j$;
    Update the PPO network by maximizing the objective function:
    $\theta_{j+1,i} = \arg\max_{\theta_i} \frac{1}{T} \sum_{j=0}^{T} \left( L_j^{\mathrm{Clip}} - c_1 L_j^{\mathrm{CR}} + c_2 E_j \right)$;
  **end for**
**end for**

As the local models are inherited from the global model, the network structure, the observation state space, the action, and the reward are defined the same as Section III. Considering that the $i$-th UE interacts with the environment at adapting epoch $j$, i.e., observes the state $o_{j,i}$, and takes action according to current policy $\pi(\theta_{j,i})$. Then the $i$-th UE receives the reward $r_j$ and observes the next state $o_{j+1,i}$. The transition $e_{j,i} = (o_{j,i}, a_{j,i}, r_j, o_{j+1,i})$ is stored in its local memory $\mathcal{M}_i$ which can be sampled in the batch to train the local models. The advantage is estimated using the GAE method and the loss function is the same as the meta-training process. The details of the meta-adapting process are described in **Algorithm 2**.

According to the definition in [8], the goal of meta-learning is to enable models to generalize well across a wide range of tasks, even those they haven't encountered during the meta-training phase. In the meta-training process, the models take the experience from different tasks with different cluster sizes. For the meta-adapting process, the models are deployed in scenarios where the number of UEs is not seen during the meta-training process, i.e., $I \notin I_k$. Moreover, the models are adapted in different scenarios with different environment settings to further verify the adaptation capability of the proposed framework, i.e., we expand the definition of the new tasks to the resource allocation in new scenarios, which will be presented in detail in Section IV.

### *E. Global Averaging of Local Models*

Unlike the meta-training process that the BS uses the centralized replay memory that collects from all UEs to train the global model, the local UEs can only access their local memories during the meta-adaptation process, which affects the robustness of the local models when encountering unfamiliar scenarios. To enable the individual models at each UE can be benefited from other UEs, the federated learning technique can be applied.

The local model is averaged to a global model, then the global model is broadcast to UEs and the UEs will continue to train the new global model locally. By averaging the models, each UE is able to benefit from the experience of other UEs, since the weights direct correspond to the experience and memory. Mathematically, the model averaging process at the central BS can be denoted as

$$\overline{\boldsymbol{W}} = \frac{\sum_{i=1}^{I} |\mathcal{B}_i| \boldsymbol{W}_i}{\sum_{i=1}^{I} |\mathcal{B}_i|}, \tag{19}$$

where $|\mathcal{B}_i|$ represents the number of number of elements in $\mathcal{B}_i$. The average algorithm shows that the averaged model will learn more from the model with more training cases. However, in the proposed *MFRL* framework, we assume that UEs share the team stage reward, which means the replay memory of each UE has an equivalent size. To ensure that the averaged model can benefit from the model that caters to the needs of QoS, we further revised the averaging algorithm that considers the success rate, which is denoted by

$$\overline{\hat{\boldsymbol{W}}} = \frac{\sum_{i=1}^{I} \eta_i \boldsymbol{W}_i}{\sum_{i=1}^{I} \eta_i}, \tag{20}$$

where $\eta_i$ is the resource allocation success rate for $UE_i$ as defined in Section II.

## IV. Numerical Results

We consider a communication scenario underlying a single cellular network. For the meta-training process, we adopt the urban micro (street canyon) scenario in [3]. For the meta-adaptation process, the pre-trained models are trained and fine-tuned in the indoor scenario, the urban macro scenario, and the rural macro scenario. The scenarios differ from each other in terms of the cell size, the path loss characteristics, the noise power spectral density, and the height of the BS antennas. The cell sizes for the indoor, urban micro (street canyon), urban macro, and rural macro scenarios are set to $25\mathrm{m} \times 25\mathrm{m}$, $100\mathrm{m} \times 100\mathrm{m}$, $500\mathrm{m} \times 500\mathrm{m}$, and $1000\mathrm{m} \times 1000\mathrm{m}$, respectively. For all of the scenarios, the BS is fixed at the center of the considered square. The heights of the BS antennas are set to $5\mathrm{m}$, $10\mathrm{m}$, $25\mathrm{m}$, and $35\mathrm{m}$, respectively. The shadowing factors are set to 8.29, 7.82, 7.8, and 8, respectively. We also adopt the simulation assumptions in [3] to model the channels, and the rest of the parameters of the proposed simulation environment are listed in Table IV.

To enable the mobility of UEs, we assume that the UEs can move with the speed from 0 meters per second $(\mathrm{m/s})$ to 1 m/s within the square. Each subcarrier has $\Delta f = 2^{\psi} \cdot 15$ kHz spacing, where $\psi$ denotes an integer. A resource block usually consists of 12 consecutive subcarriers [2], hence we set the bandwidth set of the subchannels as [0.18, 0.18, 0.36, 0.36, 0.36, 0.72, 0.72, 0.72, 1.44, 1.44] $\mathrm{MHz}$.

The network structure of local models is shown in Fig. 2. The state information is fed in two fully connected feed-forward hidden layers, which contain 512 and 256 neurons respectively. Then the PPO network diverges to actor networks and critic networks. The actor branch contains two layers for channel choice and power optimization independently, while the critic branch includes an additional hidden layer with 128 neurons, following which is the value layer for estimating the advantage function for the output of the actor network. The meta-training rate for different number of users is $5\mathrm{e}^{-7}$,

TABLE I
ENVIRONMENT PARAMETERS

| Parameter | Value |
|---|---|
| Antenna gain of the BS | 8dB |
| Antenna gain of the UEs | 3dB |
| Noise figure at the BS | 5dB |
| Noise figure at the UEs | 9dB |
| Number of UEs $I$ | 6 |
| Number of UEs for different tasks in meta-learning | [2, 4, 8] |
| Number of subchannels $N$ | 10 |
| Height of antenna of the UEs | 1.5m |
| Number of subcarriers in a RB $K$ | 12 |
| Carrier frequency $f_n, \forall n \in N$ | 6GHz |
| Cellular transmit power range | $[0, 24]$dBm |
| Static and circuit power $p_i^c, \forall i \in I$ | 18dBm |
| Minimum SINR requirements for BS $\gamma_{min}^C$ | 5 dB |
| Noise power spectral density of indoor scenario | -160 dBm/Hz |
| Noise power spectral density of urban micro scenario | -170 dBm/Hz |
| Noise power spectral density of urban macro scenario | -180 dBm/Hz |
| Noise power spectral density of rural macro scenario | -185 dBm/Hz |
| Shadowing distribution | Log-normal |
| Pathloss and shadowing update | Every 100ms |
| Fast fading update | Every 1ms |

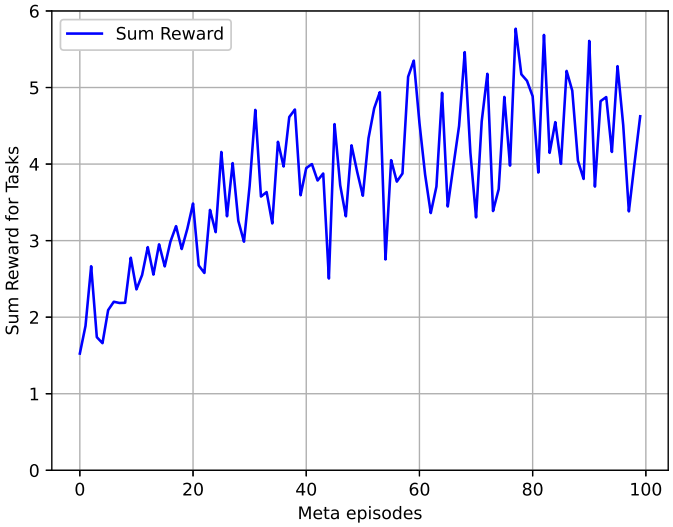



Fig. 3. Meta-training reward over the meta-training episodes. The curve represents the sum reward the agent gets from different tasks.

while the learning rate for meta adaptation is $1\mathrm{e}^{-6}$. The meta-learning rate is set relatively small to avoid the overfitting of the meta model for some specific tasks. The weight for the loss of the value function $c_1$ and entropy $c_2$ are set as 0.5 and 0.01, respectively. The sample batch size is 256, and the discount rate for the future reward $\xi$ is set to 0.9. The discount factor for the advantage function $\lambda = 0.98$ in Eq. (11) is set according to [33].

To verify the performance of the proposed *MFRL* framework, we set an ablation study and compare the performance with the following benchmarks. Note that the network structure, the advantage function estimation algorithm, and reinforcement learning related parameters are the same for all schemes.

1) **MRL**: Meta reinforcement learning benchmark. The local models are pre-trained and inherited from the global model, but the local models are not averaged by federated learning.
2) **FRL**: Federated reinforcement leanring benchmark. The local models are trained from the random initialization and averaged by the federated learning every 100 episodes.
3) **MFRL_early**: The early model of the proposed *MFRL* framework. The models are stored at half of the meta-adaptation period, i.e., at 500 episodes to evaluate the fast-adaptation performance of the proposed framework at the early stage.
4) **MARL**: The multi-agent reinforcement learning benchmark [17]. The local models are trained from random initialization and are not averaged by the federated learning technique. Each UE learns the policy according to the local observations and receives the global reward, but cannot communicate the model with the centralized cloud or other UEs.

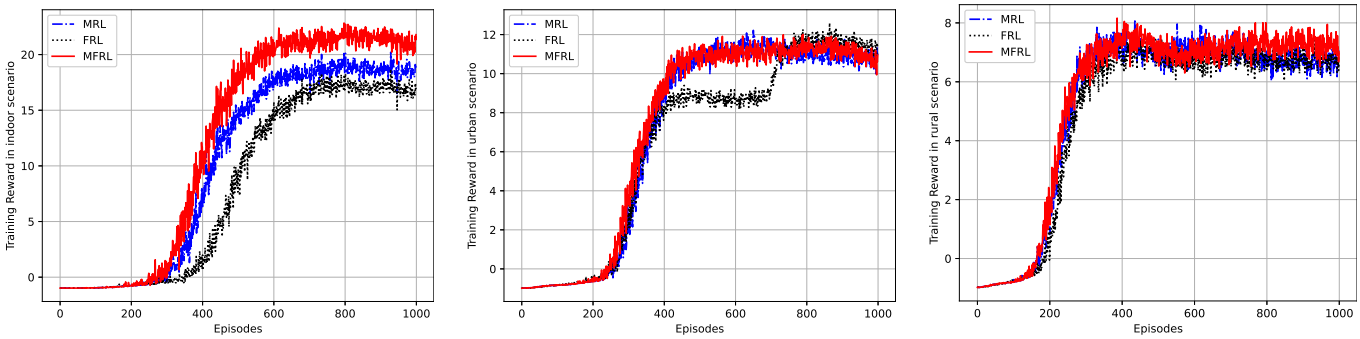
Fig. 3 demonstrates the reward for different tasks (with different amounts of users) during the meta-training process. Particularly, the meta reward is the sum of the reward of the resource allocation tasks for 2, 4, and 8 UEs in the urban micro scenario. The increase in the meta reward demonstrates the effectiveness of the meta-training. It is also noted that with the meta-training step increasing over 100 episodes, the sum reward keeps stable. This is because the meta-training process is to train a global and generalized model which can be adapted to different tasks, but the performance of the generalized model itself cannot be as well as the models for the specific tasks.

Fig. 4 shows the training reward comparison over different episodes of meta-training, from which we can see that the meta-training could lead to faster convergence and higher rewards. Since the local UEs are assigned with the same initialization, the conflict may exist during the channel assignment. Hence, at the start of the training stage, we can observe a low and negative reward until 200 to 300 episodes in all schemes due to the punishment. Nevertheless, with the execution of the training progress, we can see the lower policy entropy of the proposed scheme, which reveals the stable policy and faster convergence performance. The proposed algorithms with meta-learning can achieve faster convergence and higher training rewards, while the conventional benchmark needs more iterations to find the appropriate actions to converge. Meanwhile, the better generalization capability of the proposed *MFRL* framework brings higher system EE, verifying the fast adaptation by the meta-learning is robust to different scenarios.

To further verify the robustness of the trained local models, we set different simulation settings under each scenario. At each random testing user distribution, the system EE is averaged by 100 testing steps with fast-fading channel updates. Fig. 5 illustrates the testing performance for 10 random user distributions. It is noted that the UEs are randomly distributed in the square range. Compared to the indoor scenario, the distributions of UEs in urban and rural scenarios vary more significantly, leading to greater differences in terms of large-scale fading and path loss. The EE is unstable for different UE distributions, and we use average EE to reveal

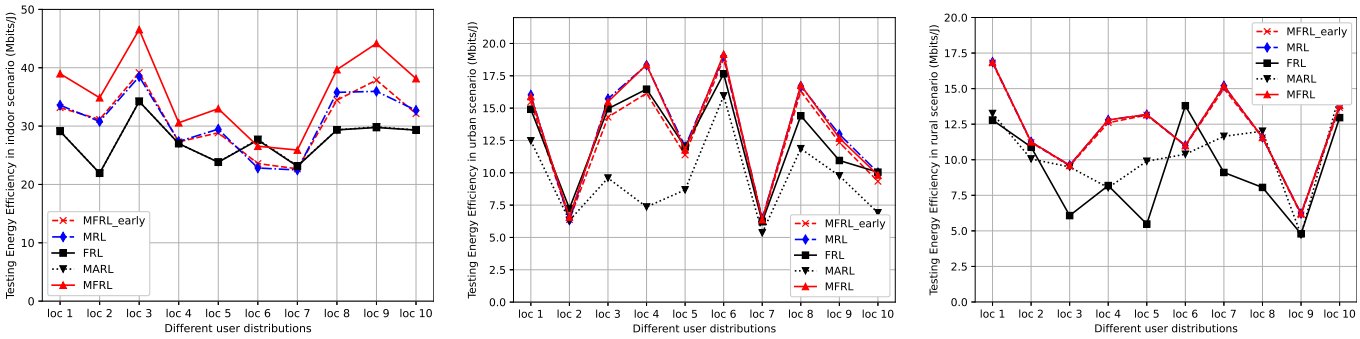
This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2023.3345363






(a) Indoor scenario. (b) Urban macro scenario. (c) Rural macro scenario.

Fig. 4. Training performance comparison of the proposed algorithm and benchmarks in three different scenarios.




(a) Indoor scenario. (b) Urban macro scenario. (c) Rural macro scenario.

Fig. 5. Testing snapshots of the proposed algorithm and benchmarks in three different scenarios.

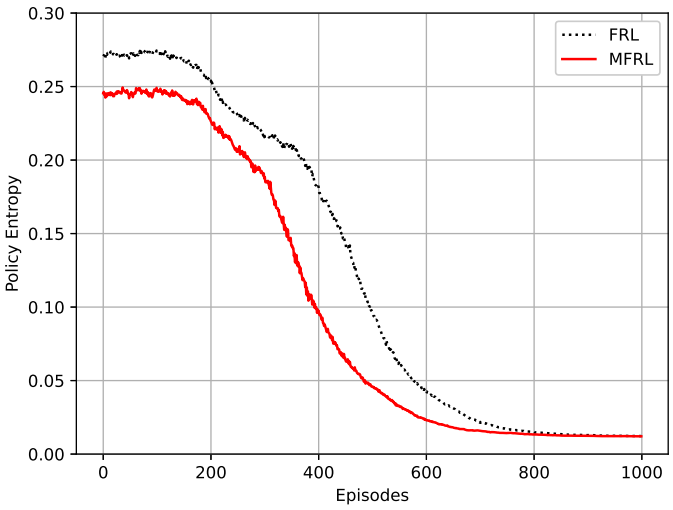



Fig. 6. Policy entropy of the *MFRL* and FRL schemes in the indoor scenario.

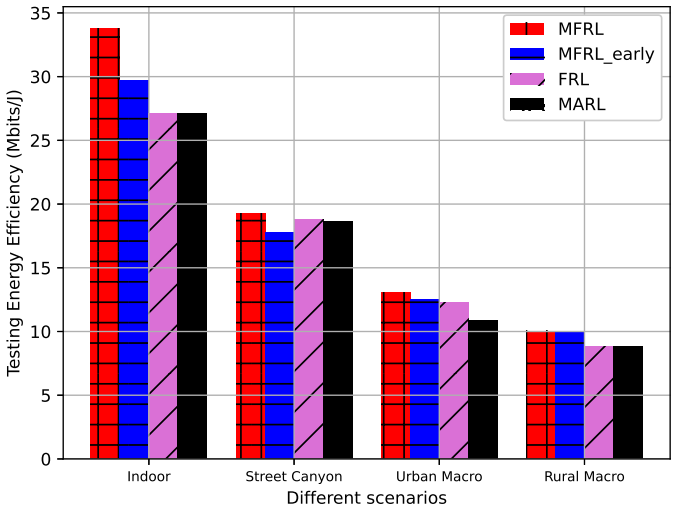



Fig. 7. Testing averaged EE performance of 100 random user distributions over the number of model averaging times.

the robustness of the algorithms. Under these circumstances, the proposed algorithm still outperforms other reinforcement learning benchmarks in terms of average system EE. We also store the local models at 500 episodes to test the performance of the algorithms at the early training stage. As expected, the proposed *MFRL* framework outperforms the MRL and FRL algorithms. Moreover, even if MFRL_early models are only trained half of the whole training period, they still provide good performances compared with the models that are not pre-trained, which verifies the fast adaptation ascendancy of the meta-learning.

To evaluate the convergence speed and the stability of the policy, and verify the fast adaptation performance of the proposed *MFRL* framework, we use the policy entropy as the measure. The policy entropy is an dimensionless index in policy gradient based reinforcement learning algorithms, to measure the randomness of a policy. As shown in Fig. 6,

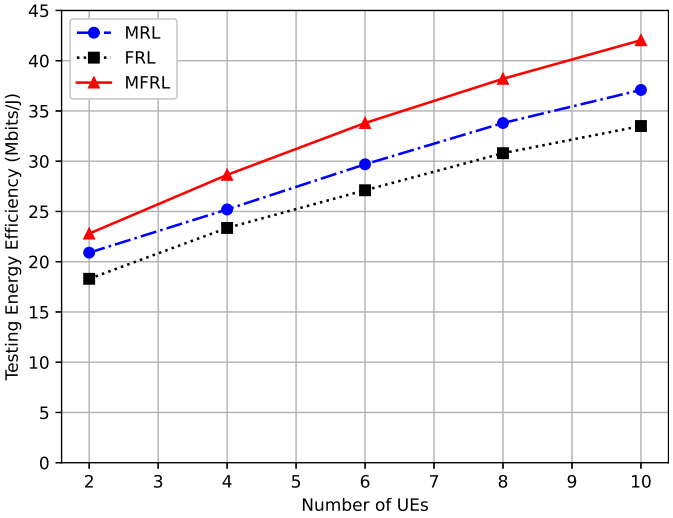



Fig. 8. Testing energy efficiency over the different number of users.

the lower entropy of the *MFRL* algorithm verifies that meta-learning can speed up the training process and achieve convergence earlier. It can be observed from Fig. 6 that the policy entropy is decreased and remains stable from 500 episodes, which represents that the policy is stable from around 500 episodes. The *MFRL* framework also achieves a similar lower entropy and faster convergence compared with the benchmarks in other scenarios, and the results are omitted due to space limitations.

Fig. 7 concludes the sum EE in different scenarios. The results are averaged according to 100 random user distributions. It is clear that the proposed *MFRL* framework achieves the highest sum EE in all of the scenarios, which verifies the robustness of the proposed scheme. Additionally, although the models for the MFRL_early benchmarks are trained half of the whole adapting period, they still achieve better performance compared with the FRL and MARL models. The *MFRL* framework and the FRL scheme enable the UEs to cooperate with each other and benefit the local models, hence also improving the overall system EE.

Fig. 8 shows the testing sum EE of the system over a different number of users. Note that for different users, the training parameters may differ slightly for the best performance. It is obvious that as the number of UEs increases, more subchannels can be accessed and the sum system EE can be improved. However, the improvement slows down as the number of UEs increases, since the bandwidth of subchannels in the proposed scenario is not equal, and when the number of UEs is less than the subchannels, it would access the subchannel with larger bandwidth for higher EE.

## V. Conclusion

In this paper, a distributed energy-efficient resource allocation scheme was developed. The system energy efficiency was maximized by jointly optimizing the channel assignment and the transmit power of user equipments. The formulated non-convex problem was solved by the proposed robust meta federated reinforcement learning framework to overcome the challenge of the computational complexity at the base station and the transmission cost by the local data. Quantity analysis and numerical results showed that the meta training model has good generalization ability under different scenarios, even if the scenarios and tasks are different. Meanwhile, the combination of federated learning and meta-learning with reinforcement learning enables the decentralized algorithm a better performance on convergence and robustness.

## References

[1] Z. Ji and Z. Qin, "Federated learning for distributed energy-efficient resource allocation," in *Proc. IEEE Int. Conf. on Commun.*, Seoul, Republic of Korea, May 2022, pp. 1–6.

[2] *Technical Specification Group Radio Access Network; NR; Physical channels and modulation; (Release 16)*, document 3GPP TS 38.211 V16.6.0, 3rd Generation Partnership Project, Jun. 2021.

[3] *Technical Specification Group Radio Access Network; Study on channel model for frequencies from 0.5 to 100 GHz (Release 16)*, 3GPP TR 38.901 V16.1.0, 3rd Generation Partnership Project, Dec. 2019.

[4] M. Rasti, S. K. Taskou, H. Tabassum, and E. Hossain, "Evolution toward 6g multi-band wireless networks: A resource management perspective," *IEEE Wireless Commun.*, vol. 29, no. 4, pp. 118–125, Aug. 2022.

[5] L. Liang, G. Y. Li, and W. Xu, "Resource allocation for D2D-enabled vehicular communications," *IEEE Trans. on Commun.*, vol. 65, no. 7, pp. 3186–3197, Apr. 2017.

[6] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn," *arXiv preprint arXiv: 1611.05763*, Jan. 2016.

[7] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rl$^2$: Fast reinforcement learning via slow reinforcement learning," *arXiv preprint arXiv: 1611.02779*, Nov. 2016.

[8] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv: 1703.03400*, Mar. 2017.

[9] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, Apr. 2020.

[10] G. Song and Y. Li, "Utility-based resource allocation and scheduling in OFDM-based wireless broadband networks," *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 127–134, Dec. 2005.

[11] M. Robat Mili, P. Tehrani, and M. Bennis, "Energy-efficient power allocation in OFDMA D2D communication by multiobjective optimization," *IEEE Wireless. Commun. Lett.*, vol. 5, no. 6, pp. 668–671, Dec. 2016.

[12] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Y. Li, "Resource allocation for text semantic communications," *IEEE Wireless Commun. Lett.*, vol. 11, no. 7, pp. 1394–1398, Jul. 2022.

[13] F. Meshkati, H. V. Poor, and S. C. Schwartz, "Energy-efficient resource allocation in wireless networks," *IEEE Signal Processing Mag.*, vol. 24, no. 3, pp. 58–68, May 2007.

[14] L. Yang, D. Wu, C. Yue, Y. Zhang, and Y. Wu, "Pricing-based channel selection for D2D content sharing in dynamic environments," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2175–2189, Dec. 2021.

[15] S. Dominic and L. Jacob, "Distributed resource allocation for D2D communications underlaying cellular networks in time-varying environment," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 388–391, Nov. 2018.

[16] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Ye Li, "Qoe-aware resource allocation for semantic communication networks," in *Proc. IEEE Glob. Commun. Conf.*, Rio de Janeiro, Brazil, Dec. 2022, pp. 3272–3277.

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

[18] L. Wang, H. Ye, L. Liang, and G. Y. Li, "Learn to compress CSI and allocate resources in vehicular networks," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3640–3653, Mar. 2020.

[19] Z. Ji, Z. Qin, and C. G. Parini, "Reconfigurable intelligent surface aided cellular networks with device-to-device users," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1808–1819, Jan. 2022.

[20] Z. Ji and Z. Qin, "Energy-efficient task offloading for semantic-aware networks," in *Proc. IEEE Int. Conf. on Commun.*, Rome, Italy, Oct. 2023, pp. 3584–3589.

[21] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Feb. 2019.

[22] Y. Yuan, G. Zheng, K.-K. Wong, and K. B. Letaief, "Meta-reinforcement learning based resource allocation for dynamic V2X communications," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8964–8977, Jul. 2021.

[23] M. He, Y. Li, X. Wang, and Z. Liu, "NOMA resource allocation method in IoV based on prioritized DQN-DDPG network," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1, pp. 1–17, Dec. 2021.

[24] J. Foerster *et al.*, "Stabilising experience replay for deep multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2017, pp. 1146–1155.

[25] Z. Qin, G. Ye Li, and H. Ye, "Federated learning and wireless communications," *IEEE Wireless Commun.*, pp. 1–7, Sep. 2021.

[26] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv: 1811.03604*, Feb. 2019.

[27] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," *arXiv preprint arXiv: 1902.01046*, Mar. 2019.

[28] H. H. Zhuo, W. Feng, Y. Lin, Q. Xu, and Q. Yang, "Federated deep reinforcement learning," *arXiv preprint arXiv: 1901.08277*, Feb. 2020.

[29] L. Zhang, H. Yin, Z. Zhou, S. Roy, and Y. Sun, "Enhancing WiFi multiple access performance with federated deep reinforcement learning," in *Proc. IEEE Veh. Technol. Conf.*, Nov. 2020, pp. 1–6.

[30] R. Zhong, X. Liu, Y. Liu, Y. Chen, and Z. Han, "Mobile reconfigurable intelligent surfaces for NOMA networks: Federated learning approaches," *arXiv preprint arXiv:2105.09462*, Mar. 2021.

[31] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., Dec. 2020, pp. 3557–3568.

[32] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, Mar. 1955.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv: 1707.06347*, Jul. 2017.

[34] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv: 1506.02438*, Jun. 2015.

[35] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.