Multi-agent Reinforcement Learning based Distributed Channel Access for Industrial Edge-Cloud Web 3.0

Chen Yang, *Member, IEEE*, Yushi Wang, Shulin Lan, *Member, IEEE*, and Liehuang Zhu, *Senior Member, IEEE*

Abstract—In the emerging Web 3.0 applications for mass customized and personalized manufacturing, smart mobile resources need to interact with each other and other resources to achieve efficient collaborative manufacturing. Existing wireless communication solutions cannot leverage multi-antenna technology and the movement direction of smart mobile resources to meet the high requirements for communication rate and reliability in high-performance manufacturing processes. Therefore, this paper proposes a task-aware distributed channel access scheme for multi-antenna smart mobile resources in a factory. First, this paper introduces an edge-cloud collaboration framework for smart factories to support autonomous wireless access point selection for mobile resources. Second, a user-centric active wireless channel access scheme is proposed and a channel resource allocation optimization problem is formulated for mobile resources to leverage multiple antennas and movement direction to address the unstable connection problem. Third, a centralized-training-and-distributed-execution multi-agent reinforcement learning (MARL) model with a specially designed neural network architecture is built for smart mobile resources, effectively using important input information of the next interaction objects for mobile resources. Simulation results show that the proposed MARL scheme outperforms common schemes of 3GPP LTE, traditional reinforcement learning schemes, and random selection schemes in improving communication rate and stability.

Index Terms—smart factory, channel access, multi-agent reinforcement learning, edge-cloud collaboration, edge computing.

1 Introduction

WEB 3.0 as the third-generation Internet, places a strong emphasis on decentralized applications and enhanced security and use machine learning and AI to empower a more intelligent and adaptive web. In the emerging industrial Web 3.0 applications, there are still challenges on how to adaptively optimize network resources. For manufacturing industry, customization and personalization products (CPPs) gradually become the core competitiveness to stabilize and expand the market. This manufacturing paradigm can meet the diverse and individual requirements [1] and at the same time requires multi-variety, small batch and ultra-short-cycle lean production for CPPs. We do not discuss the security issues in this paper and leave this as future work.

We envision a future smart factory where production resources are equipped with advanced computation and communication modules to interact and collaborate in an ultra-flexible manner during the manufacturing of CPPs. CPPs have different configurations and thus undergo distinct processing processes. To efficiently fulfill CPP orders within a short cycle and in a cost-effective manner, smart mobile resources for processing, assembly or transportation must be shared and rapidly adjusted. This allows them to interact with smart parts, each other and other resources, enabling collaborative differentiated processing or assembly, thereby achieving efficient collaborative manufacturing. Their flexible organization is realized through the reconfiguration of production machines, collaboration relationships and/or production line layout. This has the potential to significantly increase resource utilization rates, production speed, and reduce costs. Multi-antenna mobile manufacturing *things* with wireless communication and interaction needs are called agents in this paper. Agents heavily rely on wireless communication to rapidly update, reconfigure, freely move, and achieve flexible collaboration. Complex, precise, and timely control and collaboration in the customized and personalized production impose high requirements on the data transmission rate and stability between agents. Existing wireless communication solutions cannot leverage multi-antenna and the movement direction of agents to meet their requirements for communication QoS. The wireless communication QoS can be measured by communication speed and stability. Communication stability is reflected in the times that the agent disconnected from the AP and the duration that the agent remains uncon-

This work was supported by the National Key R&D Program of China under Grant 2021YFB1715700; in part by the National Natural Science Foundation of China under Grant 62103046, 72201266, 72192843, and 72192844; in part by the Fundamental Research Funds for the Central Universities under Grant E1E40805X2 and 2023CX01020. (Corresponding author: Chen Yang, Shulin Lan.)

- *Chen Yang and Liehuang Zhu is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: yangchen666@bit.edu.cn; liehuangz@bit.edu.cn).*
- *Yushi Wang is with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: awabu20@163.com).*
- *Shulin Lan is with the School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100081, China (e-mail: lanshulin@ucas.ac.cn).*

nected within a certain period of time.

Existing reinforcement learning schemes can be divided into two categories: single-agent reinforcement learning (SARL) and multi-agent reinforcement learning (MARL). SARL-based schemes are more suitable for small scale scenarios with a constant number of agents. In this case, SARL has a low complexity and can converge. However, the number of mobile agents in smart factories is often large and undergoes changes during the production process. This is due to the movement of smart parts to be processed or assembled entering and routing within the factory, while finished products will be shipped out of the facility. MARL-based schemes can handle scenarios with a large number of agents. A reinforcement learning model is established for each agent, and a decision that benefits the overall situation is made based only on the agent's perception data. Therefore, an increasing number of wireless communication studies use MARL. However, existing MARL-based schemes are often not suitable for smart CPP factories due to three main problems.

Limitations of single access based wireless communication schemes. Many MARL-based studies assume that an agent can access a single base station at one time. Naderializadeh et al. [2] proposes a distributed user scheduling and downlink power control scheme for multi-cell wireless networks. Zhang et al. [3] proposes a channel access scheme for underwater wireless optical communication networks. Park et al. [4] proposes a drone aerial base station throughput optimization scheme using the Multi-Agent Proximal Policy Optimization algorithm [5]. Kang et al. [6] proposes a scheme for a hierarchical aerial computing system composed of a high-altitude platform and multiple unmanned aerial vehicles by allocating spectrum, computing, and caching resources to increase the amount of computation tasks. Jia et al. [7] proposes a mobile edge computing task offloading strategy for a vehicular network system with limited computing resources.

The above studies adopt single access or one-to-one communication link schemes to increase the throughput of wireless communication systems, reduce the total energy consumption, and improve communication stability. However, in the smart factory, the number of agents is large and irregularly distributed, especially when production tasks are being processed at different stages in different shopfloor sites, the distribution of agents may change significantly. Therefore, a single access method may cause a problem of unstable QoS of the communication channel due to insufficient resources of the accessed base station. Therefore, the single access method cannot meet the high requirements for communication stability in the smart factory. Multi-antenna technology can enhance wireless communication speed and stability through the establishment of multiple connections.

The use of passive wireless access cannot fully meet user requirements for communication stability. Many researchers propose wireless resource allocation solutions considering the interference situation, signal-to-noise ratio, uplink and downlink communication power, and other information of the communication channel. However, users cannot actively choose the access scheme according to their current or future communication needs. Nasir et al. [8] and Lu et al. [9] propose wireless communication schemes based on deep Q-learning [10] and multi-agent deep reinforcement learning algorithms, respectively, to optimize access strategies by considering interference, signal-to-noise ratio, agent transmission power, and access point (AP) position, minimizing transmission delay. Wongphatcharatham et al. [11] proposes a scheme to improve the signal-to-noise ratio by optimizing the sending power of the base station for wireless communication channels with interference. Dutta et al. [12] proposes an adaptive MAC layer transmission strategy for heterogeneous wireless network traffic to improve wireless network throughput. Tamba et al. [13] and Lyu et al. [14] propose optimal wireless communication area coverage for wireless sensor networks composed of multiple drones as base stations, aiming to maximize the total coverage area and minimize the coverage overlap between different drones by planning the flight routes for each drone. This is based on the concept of aerial remote sensing [15].

The above studies can improve the wireless communication capacity to some extent, such as increasing communication speed, reducing communication delay, or increasing network coverage, and can meet the needs of many real-world scenarios. However, high-precision and fast production and processing in smart factories for CPPs have stricter requirements for communication speed and stability. Therefore, more proactive access solutions need to be explored considering the channel information and the needs of agents.

The general MARL method in current wireless communication area mostly hinders the utilization of key input information. Iturria-Rivera et al. [16] and Zhang et al. [17] propose load balancing and resource allocation schemes for wireless self-organizing networks respectively. Ding et al. [18] proposes a dynamic spectrum aggregation and access scheme for limited spectrum resources, using the MARL method based on the maximum entropy [19] to aggregate discrete idle channels into frequency segments and assign them to users with dynamic spectrum access (DSA) technology [20]. Ibrahim et al. [21] compares the performance differences between centralized and distributed multi-agent deep reinforcement learning in DSA problems. Emami et al. [22] proposes a drone-ground sensor connection optimization scheme using multiple unmanned aerial vehicles to collect ground sensor data.

The above MARL based solutions adopt a general input processing method, which means that the perception value in agent's observation space is first normalized and uniformly processed, and then the processed results are merged as model input or the original information is directly input into the model without additional processing. The general scheme has difficulty in distinguishing the importance of the information, and the state information with different importance levels will be mixed in the observation space, which is not conducive to the decision-making and convergence of the model.

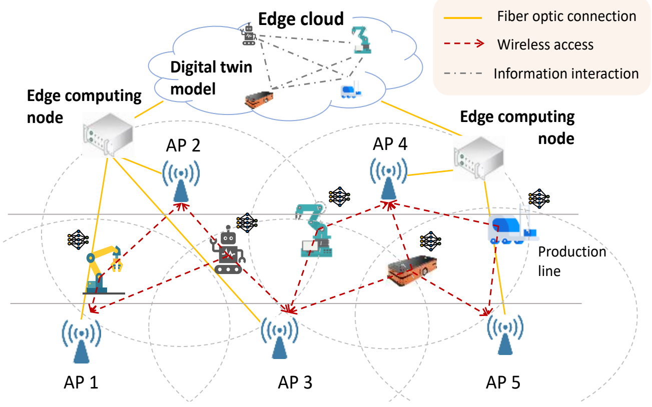

Fig. 1. Edge-Cloud Wireless Smart Factory for flexible manufacturing.

To address the above issues, we propose a task-aware active wireless network access MARL scheme for smart mobile resources in the future wireless edge-cloud factory. The agent can autonomously choose the wireless AP based on its local observation combined with its task status. Our contributions are summarized as follows:

1. We build a wireless resource allocation framework of wireless edge-cloud smart factories, which supports centralized training and distributed execution of MARL models for wireless resource allocation.

2. We propose a user-centric active wireless AP selection scheme for agents, which greatly improves communication instability in overlapping coverage areas of adjacent APs through AP selection and cooperation and increases communication performance.

3. We propose a MARL model that is trained centrally in the factory edge cloud and executed in a distributed manner on the local agents to support autonomous wireless AP selection. Considering the future communication requirements of agents, we design a novel input processing unit that includes convolutional and fully connected layers to improve the model's attention on important input information in agents' tasks, overcoming the problem of traditional reinforcement learning in fully distinguishing and utilizing critical state information of agents.

The rest of this paper is organized as follows: Section 2 introduces the system framework and formulates the optimization problem; Section 3 presents the MARL algorithm with new neural model design; Section 4 provides experimental results and analysis; and finally, Section 5 summarizes the research.

2 System Design

2.1 System Framework

To support optimal AP selection and wireless resource allocation for multi-antenna smart mobile agents, this paper proposes a smart factory framework which includes an edge cloud, edge computing nodes, and wireless access points (APs). The three kinds of components connected via fiber optic cables can collaborate to provide intelligent and high-performance wireless communication services in a smart factory.

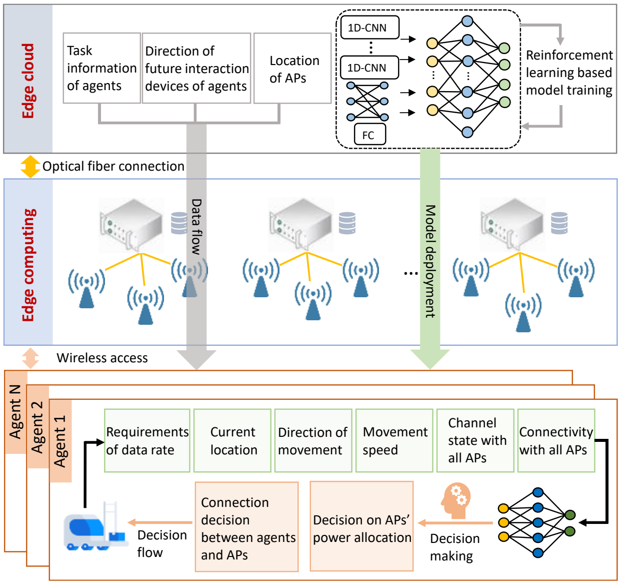
Edge cloud
Task information of agents
Direction of future interaction devices of agents
Location of APs
1D-CNN
1D-CNN
FC
Reinforcement learning based model training
Optical fiber connection
Edge computing
Data flow
Model deployment
Wireless access
Agent N
Agent 2
Agent 1
Requirements of data rate
Current location
Direction of movement
Movement speed
Channel state with all APs
Connectivity with all APs
Connection decision between agents and APs
Decision on APs' power allocation
Decision flow
Decision making

Fig. 2. Edge-Cloud Collaborative Framework for Industrial Web 3.0.

The edge cloud, the most powerful computing component in the factory, is responsible for managing edge computing nodes and connected agents, training and collaboratively executing reinforcement learning models. In this study, we assume that the fiber optic network has sufficient bandwidth to support the ultra-low delay industrial communication. The layout of the smart factory is shown in Fig. 1.

Since the number of agents can change due to the arrival of new parts or the departure of finished products, this article considers the changes in the number of agents when designing the wireless resource allocation scheme.

The agents use their sensors to collect real-time data, including channel state information, communication rate requirements, agent position, direction, speed, etc. The data can be stored locally on the agent or uploaded to edge nodes/clouds.

Agents obtain their task information and location information of APs from the edge cloud and make the decision of wireless channel access based on real-time perception data and the optimization model obtained from the edge cloud, as shown in Fig. 2.

In our scheme, the virtual AP cluster for an agent, consists of a set of accessible adjacent APs. For example, the virtual AP cluster for the black agent in Figure 1 includes AP 1, AP 2, and AP 3. The virtual AP clusters may contain duplicate APs. Each agent can select one or more APs (with the constraints of the antenna number) in its virtual cluster to meet its communication needs. The APs in a virtual cluster can use the Coordinated Multi-Point (CoMP) transmission [23] to provide improved communication services with higher rate to the agent. The multi-antenna agent can communicate cooperatively with multiple APs at the same time.

TABLE 1. Notation used in our formulation

V	agents
M	AP clusters

$q_n(t)$	the communication rate requirements of agent v_n at time t
$\Phi_v(t)$	the direction of next interaction objects agent v needs to collaborate to fulfill the task at time t
$l_{mv}(t)$	the connection status between AP m and agent v at time t
$l_{mv}^{'}(t-1,t)$	the change in the connection status between AP m and agent v from time $t-1$ to time t
$P_m^v(t)$	the downlink power assigned by AP m to agent v at time t
$U_m^v(t)$	the downlink link rate between AP m and agent v
$g_{k\to n}^t$	the independent channel gain from AP m_k to agent v_n at time t
$h_{k\to n}^t$	small-scale fading
$\vartheta_{k\to n}^t$	large-scale fading
$\gamma_{k,n}^t$	the signal-to-interference-plus-noise ratio from m_k to v_n
σ^2	the power spectral density of additive white Gaussian noise
p_k^{tr}	the transmit power
$C_{k,n}^t$	the wireless communication data rate
$B_{k,n}$	the bandwidth allocated by AP m_k to agent v_n
$s_n(t)$	the state space
$a_n(t)$	the action space
$Q_{t,v}$	penalty
$r_{t,v}$	the reward function
θ	the neural network parameters
$\pi_\theta(s,a)$	the policy function
τ	trajectory
$A_{\pi_\theta}(s,a)$	the Advantage Function
$J(\theta)$	target function
$J_t^{CLIP}(\theta)$	the clipped surrogate objective
$\sigma_t(\theta)$	the ratio between the new and old policies
μ	hyperparameter
J_t^{PPO}	the objective function of the PPO algorithm
$L_t^{VF}(\theta)$	the value function error
$E_{\pi_\theta}(s_t)$	the entropy bonus
$W_\omega(s,a)$	the joint value function
$\{\widehat{W}^V(s_t,a_t)\}_{t=1}^T$	the estimated joint value function
$J(\omega^{v_n})$	the loss function

2.2 Problem Formulation

Key notations are summarized in Table 1. We model the agents and AP clusters in the smart factory as $V=\{v_1,v_2,\ldots,v_N\}$ and $M=\{m_1,m_2,\ldots,m_K\}$, respectively. $q_n(t)$ denotes the communication rate requirements of agent v_n at time t. The edge cloud has the order list from customers and status information of manufacturing resources and can optimally assign the production tasks to agents in the factory. Therefore, the edge cloud will need to issue the information of assigned tasks to corresponding agents. The direction in which the next interaction objects that agent v needs to collaborate with to fulfill the task at time t is denoted as $\Phi_v(t)$.

We represent the connection status between AP m and agent v at time t as $l_{mv}(t)$, where the value 0 and 1 represent the disconnected and connected states, respectively. We use $l'_{mv}(t-1,t)$ to indicate the change in the connection status between AP m and agent v from time $t-1$ to time t, where the value 0 and 1 represent the "no disconnection" and "disconnection", respectively. The number of antennas of each agent determines the maximum number of APs it can access at the same time. The downlink power assigned by AP m to agent v at time t is represented as $P_m^v(t)$, which is a continuous variable in our study. The downlink link rate between AP m and agent v is represented as $U_m^v(t)$, which is also a continuous variable. As the uplink and downlink channel states for agents to access wireless networks are similar, we mainly consider the downlink link in the model to make the study easy to understand.

1) Wireless Channel Model

We use the communication rate between the AP and the agent as one of the criteria to measure the communication QoS, and the real-time channel gain of the agent as an important indicator of the communication rate. The independent channel gain from AP m_k to agent v_n at time t is represented by $g_{k\to n}^t=|h_{k\to n}^t|^2\vartheta_{k\to n}^t, t=1,2,\ldots,$ where $h_{k\to n}^t$ represents small-scale fading and $\vartheta_{k\to n}^t$ represents large-scale fading, and we use a block fading model to represent the channel gain, as described in reference [8].

The signal-to-interference-plus-noise ratio (SINR) from m_k to v_n is denoted as: $\gamma_{k,n}^t=\frac{g_{k\to n}^t p_k^{tr}}{\sum_{j\neq k} g_{j\to n}^t p_j^{tr}+\sigma^2}$, where σ^2 is the power spectral density of additive white Gaussian noise, and p_k^{tr} is the transmit power when m_k communicates with v_n. Thus, the wireless communication data rate between m_k and the currently connected agent v_n can be calculated as: $C_{k,n}^t=B_{k,n}\log_2(1+\gamma_{k,n}^t)$, where $B_{k,n}$ represents the bandwidth allocated by AP m_k to agent v_n.

2) Power Allocation Model of APs

Several AP power allocation schemes can be used according to the specific communication needs of agents in the factory. We assume that all smart agents have the same communication priority in the factory. The edge cloud can set the allocation scheme for each AP, based on the communication needs of the smart factory. The schemes include resource fairness, rate fairness, and proportional fairness.

Specifically, in Resource fairness, the AP evenly distributes the transmission power to all connected agents. In Rate fairness, the AP ensures that all connected agents obtain the same rate by adjusting the power allocation. Proportional fairness means that the channel quality and communication rate of all connected agents are considered when allocating power, and the proportion for agents with poor communication quality in the past is appropriately increased to satisfy the communication needs of all agents as much as possible. We use the exponentially weighted moving average algorithm to calculate the weighted average of the communication rate of each agent in the past 10 steps, and allocate higher wireless communication power to agents with lower weighted average values.

3) Optimization Problem Formulation

An agent autonomously chooses APs from its virtual AP cluster to connect based on their local perception data and future communication requirements. The overall goal of the system is to jointly optimize the AP selection of smart agents and the wireless channel resource allocation of each AP to maximize the overall network communication rate and minimize the times each agent disconnects from an AP (i.e., maximizing network communication stability). The optimization framework is shown as follows:

Find: $l_{mv}(t) \in \{0,1\}, P_m^v(t) \in [0, p_m^{max}]; (\forall m \in M, \forall v \in V)$

$$\textbf{max: } \{\textstyle\sum_{m\in M, v\in V} U_m^v(t), -\sum_{m\in M, v\in V} l_{mv}^{'}(t-1,t)\}. \tag{1}$$

$$\text{C1: } 0 \le \textstyle\sum_{l_{mv_n}(t)=1} p_m^{v_n} \le p_m^{max}; (n \in [1, N]). \tag{2}$$

$$\text{C2: } \gamma_{m,v}^t(\{P_m^v(t), l_{mv}(t), g_{m\to v}^t\}) \ge \gamma_m^{max}; \ (\forall m \in M, \forall v \in V). \tag{3}$$

$$\text{C3: } \textstyle\sum_{m_k \in M} l_{vm_k}(t) \ge 1; (\forall v \in V). \tag{4}$$

This optimization problem of complete collaboration between agents is to find the optimal association $l_{mv}(t)$ between the agents and the APs, as well as the optimal power allocation plan $P_m^v(t)$ for the APs, so as to maximize the overall communication rate of the agents and minimize the number of disconnections between the agents and APs over time, ensuring communication stability. Constraint C1 ensures that at any given time, the total downlink power allocated to agents cannot exceed the maximum transmission power of the AP. Constraint C2 ensures that the minimum SINR requirement should be met to determine the virtual AP cluster for each agent. Constraint C2 indicates that the actual SINR value depends on the power allocation of APs to an agent, the association status between APs and agents, and the independent channel gain between the AP and the agent. Constraint C3 ensures that at any given time, any agent in the smart factory must connect to at least one AP, ensuring that the agent always maintain a wireless connection with the APs to improve communication stability.

This is not a convex optimization problem and is difficult to solve in polynomial time. Reinforcement learning has significant advantages in solving non-convex optimization problems. Therefore, an improved reinforcement learning method with novel input modules will be used to solve this optimization problem.

3 Problem Solving Algorithm Based on Multi-Agent Reinforcement Learning

This article will implement a MARL based AP selection and wireless channel resource allocation approach for the cooperative agents to complete the predetermined production tasks. We assume that the agents are in a completely cooperative state. Therefore, in our scheme, each agent selects an AP selection action based on its perception data, and the reward is calculated through considering the actions of all agents to ensure the maximization of overall interests. The specific design is as follows:

3.1 State Space

In the MARL, each agent uses the local state to make decisions and the global state is leveraged to optimize the RL model during training. Therefore, the agent's state space and the global state space should be built respectively.

At time t, the perception data of agent v_n includes the current location of v_n, the channel conditions and the connection status between v_n and APs, etc. In addition, the agent can obtain all APs' position information and task execution status from the edge cloud. The task execution status includes the communication rate requirement $C_n'(t)$ and the direction of next interaction objects $\Phi_n(t)$. In our scheme, $\Phi_n(t)$ can assist in judging the agent's movement direction, thereby determining the changing trend of the relative distance between the agent and all APs to optimize the agent's AP selection.

To improve the convergence speed of the reinforcement learning algorithm, we further simplified the state space. We calculated the maximum achievable data rate $C_{k,n}^{max}(t)$ of each AP for the agent v_n in the current state by using the position of v_n and APs, and the channel conditions between v_n and APs, and introduced this information into the state space. Therefore, the state space $s_n(t)$ of the agent v_n at time t is represented as:

$$s_n(t) = [\{l_{Mv_n}(t)\}, \{C_{M,v_n}^{max}(t)\}, C_n'(t), \Phi_n(t)]. \tag{5}$$

The global state space $s(t)$ at time t is the collection of state spaces of all agents, that is:

$$s(t) = [\{l_{MV}(t)\}, \{C_{M,V}^{max}(t)\}, \{C_V'(t)\}, \{\Phi_V(t)\}]. \tag{6}$$

3.2 Action Space

We establish an action space for each agent's local decision-making and a global action space for the reinforcement learning model. Specifically, each agent autonomously selects or de-selects the APs it needs based on its current state from its virtual AP cluster. For each AP, the agent has two action choices, namely, maintaining or changing the existing connection state. After determining the APs to connect, the agent can obtain the corresponding downlink bandwidth from each connected AP by using the APs' corresponding power allocation model. Therefore, at time t, the action space of agent v_n based on state $s_n(t)$ is:

$$a_n(t) = [\{l_{Mv_n}'(t-1,t)\}, \{C_{M,v_n}^t\}]. \tag{7}$$

The global action space based on the state $s(t)$ at time t is the collection of action spaces of all agents, that is:

$$a(t) = [\{l_{MV}'(t-1,t)\}, \{C_{M,V}^t\}]. \tag{8}$$

3.3 Reward Function

The optimization goal of wireless communication in the smart factory is to achieve higher communication quality of service (QoS), i.e., maximize the overall communication rate and minimize the number of disconnections between agents and APs. To achieve the maximization of global benefits, we do not set rewards of individual action for each agent in the reinforcement learning model training process. Instead, we adopt a global reward approach, which calculates the sum of rewards obtained by all agents at the end of each time step t to measure the goodness of the decisions made by all agents at time t. Setting a global reward may cause some agents to make decisions that are not optimal for themselves, but it can maximize the overall benefits.

Specifically, each agent v will receive a higher reward for obtaining the highest possible communication rate from all connected APs, i.e., $\sum_{m\in M} C_{m,v}^{t}$. However, to avoid the situation that agents constantly try to switch to other APs in order to pursue higher communication rates, which severely affects communication stability. Therefore, we define a penalty for each agent v for disconnecting from an AP:

$$Q_{t,v} = \max\{l'_{mv}(t-1,t), 0\}. \tag{9}$$

Therefore, the reward function of agent v at time t is defined as:

$$r_{t,v} = \psi[\sum_{m\in M} C_{m,v}^{t} - \eta Q_{t,v}]. \tag{10}$$

where, η represents the penalty coefficient for each agent v disconnecting from an AP. Setting a proper value for η can effectively extend the time for an agent to remain connected to an AP. $\psi \in R^{+}$ is used to determine the range of rewards. The global reward for the system at time t is the sum of rewards obtained by all agents, i.e.,

$$r_t = \sum_{v\in V} r_{t,v}. \tag{11}$$

The global reward can guide the MARL algorithm to balance the total communication rate and agents' disconnection action by adjusting η, and achieve optimal communication QoS.

3.4 Multi-agent Reinforcement Learning Model

1) Input Processing Unit of the Model

To enhance the model's attention to important state information, we designed the preprocessing part of the MARL model's input. Current reinforcement learning solutions usually normalize the input information and then merge the processed information as the model input. However, in our scenario, these methods will mix agents' state information such as the position, velocity, and direction with environmental information such as channel conditions and maximum downlink rates, causing the critical input information to be undistinguished and not fully utilized by the model.

In our research, we found that the direction of next interaction object in the agent state information can play a critical role in optimizing wireless AP selection strategies. The goal of feature extraction can be achieved by building fully connected layers or convolutional layers in neural networks. The utilization of fully connected layers often involves larger models and more parameters, while the size of convolutional layers is much smaller. However, in practical applications, we found that when using fully connected layers and convolutional layers in the input layer of a neural network, the information input to the fully connected layers can be more effectively utilized by the model, that is, it receives more attention. Therefore, we input the movement direction information of the agent and other input information into the fully connected layers and convolutional layers in the model, respectively, thereby improving the model's attention to the movement direction information of the agent.

Specifically, we designed an input processing unit that includes convolutional layers and fully connected layers to achieve input information feature extraction. We input the environment information perceived by the agent and the agent communication rate requirement to a 1D convolutional layer and input the movement direction of the agent to a fully connected layer with 128 dimensions. In the model construction process, we found that using a fully connected layer with 128 dimensions is sufficient to obtain satisfactory convergence performance, and better convergence performance can possibly be obtained through further optimizing the model parameters.

The edge cloud is responsible for uniformly managing the specific production tasks performed by each agent and can determine the agent's movement direction based on the task information. According to the actual situation in the factory, the movement direction of each agent remains unchanged for a period of time. In the experiment, we set 1, 2, 3, and 4 to correspond to the north, east, south, and west directions of the agent.

2) Decision-Making Unit of the Model

The input pre-processing unit is connected to the fusion unit for decision-making, which includes the policy network (Actor) and the evaluation network (Critic) for each agent. The fusion unit supports centralized training and distributed execution of the MARL models, so the model has two different operating modes: in the training stage, all Actor and Critic networks are required, and the training objective is to enable the Actor network to make globally optimal decisions; in the execution stage, only the Actor network needs to be used to make decisions based on the real-time status of the agent.

Specifically, each agent has its own Actor and Critic network. In the model training stage, the Critic network of each agent will be trained based on global data to estimate the joint value function. The global data is composed of the local status of all agents and the decisions made by the Actor networks. As shown in Fig. 3, the Actor network of an agent only makes decisions based on the local status of the agent. In the model execution stage, each agent does not need global information and can choose actions that are beneficial to the global situation. In our solution, all Actor and Critic networks include a fully connected layer with 256 dimensions. The output of the Actor network is the action policy of the agent to connect or disconnect the AP. The bandwidth allocated to an agent by the AP is determined according to the power allocation scheme of the AP.

3.5 MARL Algorithm

According to our wireless communication optimization scheme, in each state $s(t)$, each agent takes action $a_n(t)$ based on its local state $s_n(t)$ and obtains a global reward r_t, and the system changes to the next state $s(t+1)$. Since the state space contains continuous variables such as achievable data rates for agents with respect to APs, we use the Proximal Policy Optimization (PPO) algorithm [24], based on the Policy Gradient (PG) algorithm [25], for improvement. PPO uses an Actor network to output action policies based on state information and a Critic net-

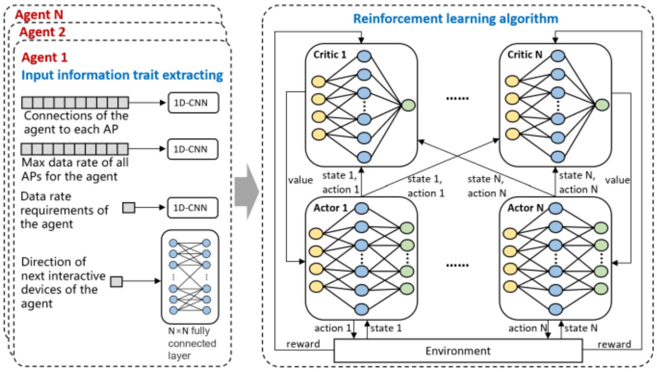
Agent N
Agent 2
Agent 1
Input information trait extracting
Connections of the agent to each AP
1D-CNN
Max data rate of all APs for the agent
1D-CNN
Data rate requirements of the agent
1D-CNN
Direction of next interactive devices of the agent
N×N fully connected layer
Reinforcement learning algorithm
Critic 1
Critic N
......
value
state 1, action 1
state 1, action 1
state N, action N
state N, action N
value
Actor 1
Actor N
......
action 1
state 1
action N
state N
reward
Environment
reward

Fig. 3. Multi-agent Reinforcement Learning with Novel Input Module.

work to provide scores for output actions, which are used to update policy parameters.

1) Training the Actor Network

In the MARL algorithm, the Actor network takes the current state information and outputs the action policy. The policy function $\pi_\theta(s,a)$ increases the probability of obtaining actions with higher rewards by updating the neural network parameters θ. In each iteration, the MARL algorithm first collects data for T time steps. During the data collection period, each agent interacts with the environment using its policy π_θ, and finally collects a trajectory $\tau = \{\{s_1,a_1\},\{s_2,a_2\},\ldots,\{s_T,a_T\}\}$ of T time steps. Then, the Advantage Function $A_{\pi_\theta}(s,a)$ is calculated, which represents the magnitude of the effect of action a in state s.

Then the algorithm uses the collected data to train and update parameters for E epochs. At this time, a target function $J(\theta)$ is constructed, whose gradient is equal to the policy gradient of π_θ. Policy updates are achieved by optimizing the target function. Unlike the policy gradient algorithm, we have improved the PPO algorithm in two aspects.

Firstly, the PPO algorithm uses the clipped surrogate objective for neural network parameter updates. The clipped surrogate objective is defined as:

$$J_t^{CLIP}(\theta) = E_t[\min(\sigma_t(\theta), clip(\sigma_t(\theta), 1-\mu, 1+\mu))\, A_t(\theta_{old})]. \quad (12)$$

where $\sigma_t(\theta)$ is the ratio between the new and old policies. When $A_t(\theta_{old}) > 0$, the action in the current state s_t receives a reward, $\sigma_t(\theta)$ increases. Otherwise, when $A_t(\theta_{old}) < 0$, the action in the current state s_t is punished, $\sigma_t(\theta)$ decreases. To avoid much large differences between the new and old policies, the PPO algorithm uses the truncation method to limit $\sigma_t(\theta)$ to $[1-\mu, 1+\mu]$, where μ is a hyperparameter.

Secondly, the PPO algorithm uses the Generalized Advantage Estimation to estimate $A_t(\theta_{old})$. That is, the advantage estimation is smoothed by considering the advantages of each step after state s_t and weighting them according to their distance from the current state. Through these improvements, the PPO algorithm has better convergence performance than the policy gradient algorithm.

The objective function of the PPO algorithm can be written as:

$$J_t^{PPO}(\theta) = E_t[J_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 E_{\pi_\theta}(s_t)]. \quad (13)$$

where $L_t^{VF}(\theta)$ is the value function error, $E_{\pi_\theta}(s_t)$ is the entropy bonus used to increase the model's uncertainty, thereby promoting the exploration of more possibilities during model training. c_1 and c_2 are coefficients.

After obtaining the objective function, we use the gradient ascent method to update all agents based on the sampled data.

2) Training Critic Network

In the MARL algorithm, the Critic network maps the global action and state into a single value using the joint value function $W_\omega(s,a)$, which is controlled by the parameter ω, and feeds the value back to the Actor network.

During the data collection phase, each agent interacts with the environment, and in each iteration, the estimated joint value function $\{\widehat{W}^V(s_t,a_t)\}_{t=1}^T$ is calculated using the T time steps of all agents' trajectories. In the training process of E epochs, ω in the Critic network of each agent is updated by minimizing the loss function $J(\omega^{v_n}) = \left(\widehat{W}^{v_n}(s_t,a_t) - W_{\omega^{v_n}}(s_t,a_t)\right)^2$.

3) MARL based Active Wireless Communication Optimization Strategy for the smart Factory

We designed a MARL algorithm based on PPO algorithm to effectively solve the problem of active AP selection and channel resource allocation in a smart factory. The algorithm is shown in Algorithm 1:

Algorithm 1: MARL based Active AP Selection for the smart Factory

```
Input: direction $\{\Phi_V(1)\}$ of agent's next interaction objects, connection status $\{l_{MV}(1)\}$ between APs and agents, current communication rate requirement $\{C_V'(1)\}$ of agents, maximum achievable data rate $\{C_{M,V}^{max}(1)\}$ of the AP for agents.
Output: agent selects to connect/disconnect from APs, action a(t).
Initialization of neural network parameters $\theta$, $\omega$;
Initialization of iteration count K=[1, k], step count T=[1, i], epoch count E=[1, e];
For iteration = 1 to k:
    Get direction $\{\Phi_V(1)\}$ of agent's next interaction objects, connection status $\{l_{MV}(1)\}$, $\{C_V'(1)\}$ from edge cloud;
    Calculate maximum achievable data rate $\{C_{M,V}^{max}(1)\}$ of AP for agent;
    Initialize state $s(1) = [\{l_{MV}(1)\}, \{C_{M,V}^{max}(1)\}, \{C_V'(1)\}, \{\Phi_V(1)\}]$;
    for T = 1 to i
        Each agent independently chooses action $\{a_n(t)\}_{n=1}^N$ according to the old strategy;
        Execute actions $\{a_n(t)\}_{n=1}^N$, obtain global reward $r_t = \sum_{v\in V} r_{t,v}$, update state $s(t+1)$;
        Calculate advantage estimation $\{A_1, A_2, \ldots, A_i\}$;
    end for
    for Epoch = 1 to e
        Calculate $J^{PPO}(\theta)$ for each agent;
        Calculate $\nabla_\theta J^{PPO}(\theta)$ for each agent;
        Update $\theta$ for each agent using gradient ascent method;
        Update $\omega$ for each agent using gradient ascent method;
        end for
```

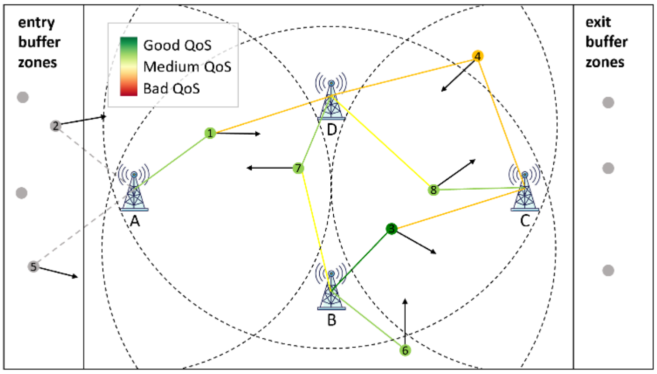

Fig. 4. Simulation Environment of Wireless Communication Scenarios in Smart Factories.

```
        Update and save $\theta$, $\omega$ for each agent;
    end for
```

Algorithm 1 is the training process for active AP selection and channel resource allocation in smart factories based on MARL algorithms. The input of the algorithm is the initialized global state (collected states of all agents), and the output is the action of each agent to connect or disconnect from the AP.

The algorithm is iterated k times, and in each iteration, all agents move i steps and update the model parameters. The time complexity of the algorithm is $O(n^2)$.

4 Numerical Results

4.1 Experimental Setting

We implemented the proposed MARL algorithm using TensorFlow 2.7 and the RLlib framework. We used a 4-core 8-thread Intel Core i7-10510U CPU and an Nvidia GeForce MX250 GPU with 16GB of memory for simulation and reinforcement learning model training.

1) Neural Network Setting

Our MARL neural network architecture designed for active wireless channel access decision-making by agents in the smart factory is shown in Fig. 3. In our approach, each agent uses the same neural network model, which consists of input processing units and fusion units for decision-making. The input processing units consist of three 1D convolutional layers and one fully connected layer. The convolutional layer filters are set to [1, 2, 1], and the ReLU activation function is employed; the fully connected layer contains 128 neurons, and tanh is used as the activation function. The fusion unit for decision-making consists of a policy network and an evaluation network, both of which have a fully connected layer with 256 neurons and use ReLU as the activation function. This model has demonstrated good convergence performance in our simulation environment, and can be optimized for other environments as per specific requirements. The output of the neural network is the action of each agent in connecting or disconnecting APs.

2) Simulation Environment

We created a simulation environment for the wireless communication scenario in the smart factory, as shown in Fig. 4. The environment spanned an area of 112 meters by 80 meters, with entry and exit buffer zones on both sides for components and products, respectively. The agents can move freely in the smart factory area and enter and exit this area through the buffer zones. We placed four APs throughout the area to ensure complete wireless coverage. The APs named A, B, C, and D, were located at coordinates (10, 40), (56, 10), (102, 40), and (56, 70), respectively. Each AP had a transmission power of 30 dBm, operated at a frequency of 2500 MHz, had a downlink bandwidth of 9 MHz, and a noise power set to 10^{-9} mW. In the simulation, we measured execution time t in terms of the number of steps taken by the agent, with one step corresponding to the completion of one move. We set 100 steps as one episode and reset t after each completed episode. During the simulation, we did not consider bandwidth consumption and communication delay between the edge cloud and agents.

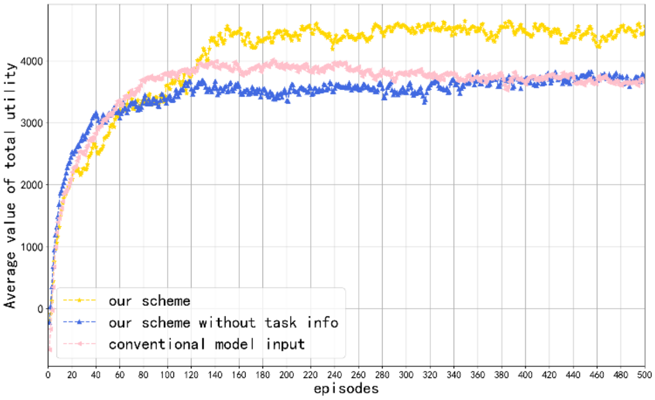

Fig. 5. Algorithm Utility and Convergence.

APs' power settings: In a smart factory, the power allocation scheme of APs typically remains unchanged once set. To reflect this reality in our simulation, we only set the power allocation mode for each AP during the initialization stage of each episode. We used the resource fair, rate fair, orthogonal fair, and resource fair power allocation schemes for APs A, B, C, and D, respectively.

Agent motion trajectory setting: In a smart factory, agents can be either static or moving. However, in our simulation, we only considered mobile agents because the communication access situation for static agents is relatively fixed and communication strategies are easier to develop. During the initialization stage of each episode, we randomly set the positions and movement directions of all agents. Moreover, to ensure that the agents remain within the simulation area throughout each episode, we assign a random speed between 0.4 meters/step and 1.2 meters/step to each agent.

3) Comparison Scheme

To verify the effectiveness of our proposed MARL scheme, we compared it with several other schemes. First, we used the 3GPP LTE common access point selection scheme as the benchmark. In addition, we designed three other schemes for comparison experiments: a reinforcement learning scheme that does not include the next step interaction direction of the agents, a reinforcement learn-

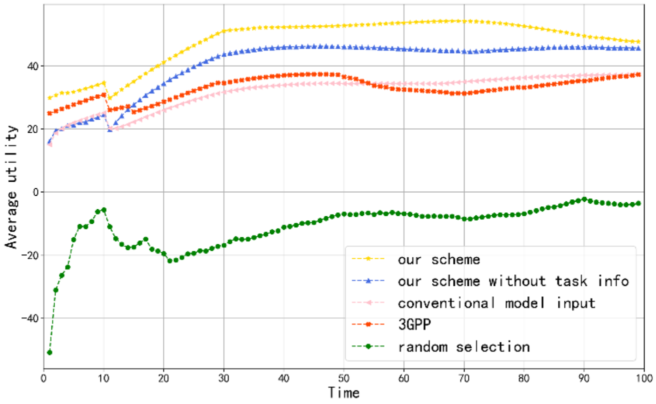

Fig. 6. Total Utility During a Test Episode.

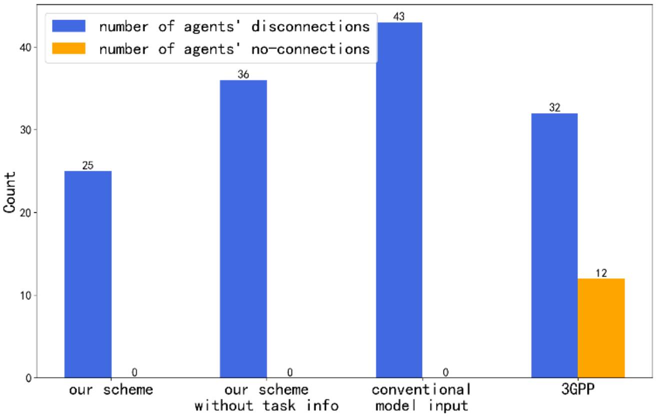

Fig. 7. Number of Disconnections and No-Connections.

ing model with the normalized input scheme, and a random AP selection scheme.

The 3GPP LTE common access point selection scheme selects the AP with the highest SINR among all APs for access, and maintains the connection until the SINR drops below the minimum threshold, at which point it disconnects and accesses another AP. To verify the significance of future interactive objects of the agent in wireless channel access policy, we designed a reinforcement learning scheme that does not consider the agent's future interactive objects, where the network architecture remains unchanged. To verify the effectiveness of the proposed network architecture, we designed a reinforcement learning model with the input normalization scheme, where all input information is normalized and then merged as the input of the reinforcement learning model. We also implemented a scheme where all agents randomly select the APs to access at each step regardless of the current channel environment and state conditions.

4.2 Results and Analysis

We conducted comparative experiments between our proposed MARL-based active wireless channel access approach and other approaches. For all the methods that require reinforcement learning, we trained the models for 500 episodes. We used the aggregate real-time communication rate of all smart agents as one of the metrics to assess the communication QoS in the experiment. Since the proposed multi-agent reinforcement learning method supports changes in the number of agents, we set the number of agents to be between 5-8 and allowed them to enter or exit the simulated area randomly in the simulation experiments.

1) Algorithm Utility and Convergence

The algorithm utility denotes the aggregate real-time communication rate of all smart agents resulting from the designed algorithm. We compared the convergence and algorithm utility of three reinforcement learning models: the proposed MARL scheme, the MARL scheme that does not include the direction of the next interaction object of the agent namely “undirected”, and the MARL model with the normalized input scheme namely “conventional model input”. We trained the model under each scheme for 500 episodes, and calculated the average value of total utilities of all agents (i.e., summarizing the utility of all the agents and then averaging the results), to observe the convergence of the model as shown in Fig. 5. All the models converged within 150 episodes, indicating that the model parameters we set were reasonable. Additionally, we set up a random scheme for agents to entry or exit, which also affects the stability of the average value of total utilities. After model convergence, the average value of total utilities of the proposed model was around 4500, while the other two schemes were between 3000 and 4000. This demonstrates that the direction of the agent’s next interaction object does play a significant role in wireless channel access decision-making, and also confirms the effectiveness of the MARL model with the input processing unit.

2) Total Utility During a Test Episode

After the completion of model training for all reinforcement learning schemes, the proposed scheme and four comparison schemes were simulated for a full episode to compare the total agent utility at each step. To ensure fairness, we set the same movement and entry/exit plan for all schemes. As shown in Fig. 6, the random selection has much lower average utility than the other four schemes, indicating that the other four schemes can optimize the wireless channel access problem. We set three times of agent entry and exit from the tenth step onwards. The average utility of the 3GPP LTE scheme fluctuates significantly each time an agent enters or exits, while the fluctuations in the three MARL schemes are much smaller. This is because the 3GPP LTE scheme uses the single-access approach. The number of agents in the environment changes, especially new agents enter the environment, easily affecting other agents connected to the same base station, and even cause poorly connected agents to disconnect. The multiple access approach can effectively alleviate this problem.

It can be seen that the total utility of the proposed scheme is better than the other four schemes, and the MARL scheme without the agent's future task information in the input information is better than the 3GPP LTE scheme and the MARL scheme with the conventional model input module. This outcome verified the effectiveness of our MARL model design and supports the idea that the future task information of agents can play a critical role in wireless channel access strategy.

3) Number of Disconnections and No-Connections

The previous two comparison experiments focused on evaluating the benefits of our proposed solution from a utility perspective. However, since the utility only reflects the communication rate, we also examined communication stability by analyzing the times the agents were disconnected from the AP and the total number of steps the agents were not connected to APs simultaneously during a complete episode simulation. A lower frequency of disconnections and no-connections indicates better communication stability. Our analysis of these metrics is presented in Fig. 7.

Our scheme differs from the 3GPP LTE scheme in that it uses CoMP technology, which allows each agent to access more than one AP simultaneously. Based on our simulation results, we found that on average, each agent accesses more than two APs at the same time in our scheme. Our proposed scheme can support more stable communication if the number of disconnections is less than twice of 3GPP LTE, because an agent in our scheme with two antennas has twice the disconnection probability than 3GPP LTE scheme. Fig. 7 shows that the number of agents' disconnections in our scheme is 25 times, while the number of disconnections is 36 times and 43 times respectively for the MARL scheme without future interaction information and the conventional MARL scheme. All of these are less than twice of 3GPP LTE scheme, which is 64 times, indicating that the disconnection penalty we set in the reward function of the MARL algorithm works well. Fig. 7 also shows that the number of disconnections of the other two MARL schemes are more than that of our scheme. This indicates that adding future interaction information in MARL can effectively improve the stability of wireless channel access services, and that the architecture design of our model is more conducive to the use of the agent's future interaction information (direction of the next interaction object).

Overall, in smart factories, high wireless communication stability and rate are essential, because mobile agents need to engage in frequent interactions and collaborations with each other or other devices to enable the efficient, collaborative, differentiated manufacturing of CPPs. Thus, it would be highly detrimental to high-performance production and AGV material transportation if an agent becomes disconnected from all APs, or collaboration orders are not timely transmitted. Such disconnection or delayed messages may cause equipment collisions, damage, or other severe impacts on production progress. Therefore, it is necessary to avoid disconnection situations and high communication delay, to well meet the demand for communication flexibility in multi-variety, small batch, and ultra-short cycle lean production. Fig. 7 shows that only the 3GPP LTE scheme has "no connectivity" situations for agents, while our proposed scheme effectively avoids this situation by properly setting the disconnection penalty parameter in MARL. Fig. 5 and Fig. 6 indicate that the proposed algorithm outperforms other methods in terms of the total communication rate.

4) Cost of Training Time

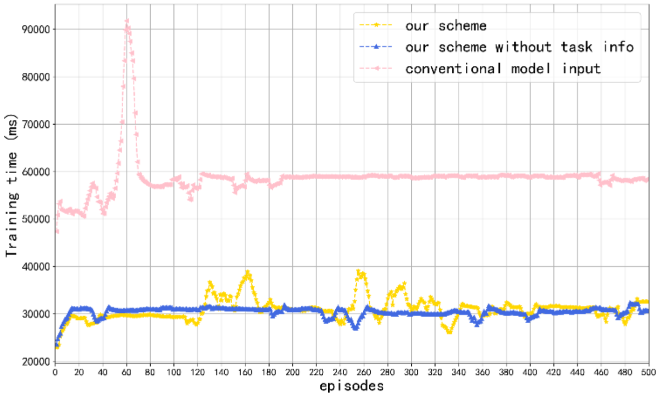

Fig. 8. Cost of Training Time.

The training time of the reinforcement learning model affects the deployment speed. Therefore, we compared the training times of our proposed scheme, the MARL scheme without the direction of the agent's next interaction object, and the conventional MARL model. Since each agent runs its own local MARL model, we calculated the total training time of all agents' MARL models (the sum of the training time for all agent models), as shown in Fig. 8. The conventional MARL model took nearly twice training time than our proposed scheme, which indicates that our preprocessing unit for the reinforcement learning model can effectively reduce the input information size and improve the efficiency of the model training. Although the MARL model we designed has higher input dimensions than the MARL scheme without the direction of the agent's next interaction object, the training time of the two models does not differ significantly. This suggests that after the input data size is reduced to a certain extent, the fusion unit for decision-making mainly affects the training time of the reinforcement learning model.

5) Running Time Costs

We compared the average running time of all the schemes during an episode, which reflects the time taken by the agents to make wireless channel access decisions and is one of the factors affecting communication delay. We used CPU to execute all the schemes, as shown in Fig. 9. The running times of the different schemes were relatively close, with the random selection scheme has the shortest running time, as it does not require a model or the execution of complex algorithms and can be considered to have the theoretical shortest time. The running time of the conventional MARL model was relatively high, as this scheme used a large input size for the reinforcement learning model. Our proposed MARL scheme had almost the same running time as the 3GPP LTE scheme, and both were slightly higher than the MARL scheme without the direction of the agent's next interaction object, indicating that the complexity of our model was reasonable, and that the input data size for the MARL model proposed has some impacts on the running time of the model.

5 Conclusion

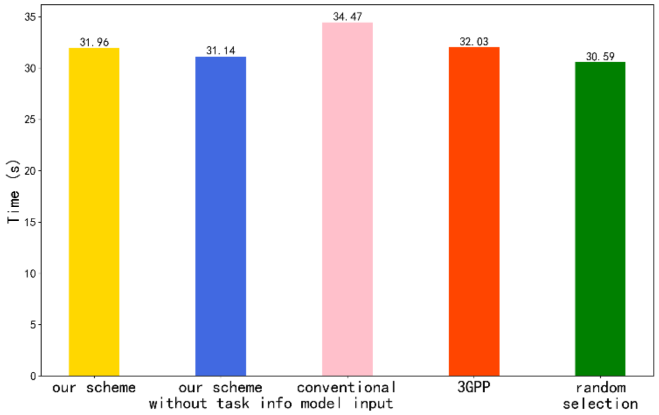

Fig. 9. Running Time Costs.

In the emerging Web 3.0 applications for the manufacturing industry, there are still challenges on how to adaptively optimize wireless network resources to meet the needs of mobile collaborative resources for customization and personalization production. In this paper, we proposed a task-aware proactive wireless channel access scheme in smart factories and proposed a multi-agent reinforcement learning method to improve QoS. We adopted an agent-centric networking scheme to enhance communication stability and realized proactive wireless channel access of agents through a centralized training and distributed execution multi-agent reinforcement learning algorithm, where the direction of the next interaction object of the agent is further introduced. Then, we improved the reinforcement learning model with a novel input information preprocessing unit to effectively improve communication QoS by processing input information with different importance. Simulation results showed that compared with traditional schemes, our proposed reinforcement learning scheme performed better in terms of communication rate and communication stability.

References

[1] C. Yang, S. L. Lan, W. Shen, G. Q. Huang, X. Wang, and T. Lin, “Towards Product Customization and Personalization in IoT-enabled Cloud Manufacturing,” *Cluster Computing*, vol. 20, no. 2, pp. 1717–1730, 2017.

[2] N. Naderializadeh, J. J. Sydir, M. Simsek, and H. Nikopour, “Resource Management in Wireless Networks via Multi-Agent Deep Reinforcement Learning,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3507–3523, Jun. 2021.

[3] Z. Zhang, L. Zhang, and Z. Chen, “Multi-Agent Reinforcement Learning Based Channel Access Scheme for Underwater Optical Wireless Communication Networks,” in *2021 15th International Symposium on Medical Information and Communication Technology (ISMICT)*. Xiamen, China: IEEE, Apr. 2021, pp. 65–69.

[4] Y. M. Park, S. S. Hassan, and C. S. Hong, “Maximizing Throughput of Aerial Base Stations via Resources-based Multi-Agent Proximal Policy Optimization: A Deep Reinforcement Learning Approach,” in *2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*. Takamatsu, Japan: IEEE, Sep. 2022, pp. 1–4.

[5] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, “The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games,” Nov. 2022.

[6] H. Kang, X. Chang, J. Misic, V. B. Misic, J. Fan, and Y. Liu, “Cooperative UAV Resource Allocation and Task Offloading in Hierarchical Aerial Computing Systems: A MAPPO Based Approach,” *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[7] Y. Jia, C. Zhang, Y. Huang, and W. Zhang, “Lyapunov Optimization Based Mobile Edge Computing for Internet of Vehicles Systems,” *IEEE Transactions on Communications*, vol. 70, no. 11, pp. 7418–7433, Nov. 2022.

[8] Y. S. Nasir and D. Guo, “Multi-Agent Deep Reinforcement Learning for Dynamic Power Allocation in Wireless Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[9] Z. Lu, C. Zhong, and M. C. Gursoy, “Dynamic Channel Access and Power Control in Wireless Interference Networks via Multi-Agent Deep Reinforcement Learning,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1588–1601, Feb. 2022.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[11] T. Wongphatcharatham, W. Phakphisut, T. Wijitpornchai, P. Areeprayoonkij, T. Jaruvitayakovit, and P. Hannanta-anan, “Multi-Agent Q-Learning for Power Allocation in Interference Channel,” in *2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*. Phuket, Thailand: IEEE, Jul. 2022, pp. 876–879.

[12] H. Dutta and S. Biswas, “Towards Multi-agent Reinforcement Learning for Wireless Network Protocol Synthesis,” in *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. Bangalore, India: IEEE, Jan. 2021, pp. 614–622.

[13] T. A. Tamba, “Optimizing the Area Coverage of Networked UAVs using Multi-Agent Reinforcement Learning,” in *2021 International Conference on Instrumentation, Control, and Automation (ICA)*. Bandung, Indonesia: IEEE, Aug. 2021, pp. 197–201.

[14] H. Lyu, S. Hwang, and H. J. Yang, “Multi-Agent Reinforcement Learning-Based Coverage Maximization for Fixed-Wing Base Stations,” in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. Jeju Island, Korea, Republic of: IEEE, Oct. 2022, pp. 1529–1532.

[15] A. Barrientos, J. Colorado, J. D. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, “Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots: Aerial Remote Sensing in Agriculture,” *Journal of Field Robotics*, vol. 28, no. 5, pp. 667–689, Sep. 2011.

[16] P. E. Iturria-Rivera and M. Erol-Kantarci, “Competitive MultiAgent Load Balancing with Adaptive Policies in Wireless Networks,” in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. Las Vegas, NV, USA: IEEE, Jan. 2022, pp. 796–801.

[17] S. Zhang, Z. Ni, L. Kuang, C. Jiang, and X. Zhao, “Load-Aware Distributed Resource Allocation for MF-TDMA Ad Hoc Networks: A Multi-Agent DRL Approach,” *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 6, pp. 4426–4443, Nov. 2022.

[18] W. Ding, W. Zhang, D. Wang, J. Sun, and C.-X. Wang, “Dynamic

Spectrum Aggregation and Access Scheme Based on MultiAgent Actor-Critic Reinforcement Learning," in *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*. Changsha, China: IEEE, Oct. 2021, pp. 1–5.

[19] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," Aug. 2018.

[20] Q. Zhao and B. Sadler, "A Survey of Dynamic Spectrum Access," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, May 2007.

[21] A. M. Ibrahim, K.-L. A. Yau, and L. M. Hong, "Implications of Centralized and Distributed Multi-Agent Deep Reinforcement Learning in Dynamic Spectrum Access," in *2022 IEEE 6th International Symposium on Telecommunication Technologies (ISTT)*. Johor Bahru, Malaysia: IEEE, Nov. 2022, pp. 62–67.

[22] Y. Emami, B. Wei, K. Li, W. Ni, and E. Tovar, "Deep Q-Networks for Aerial Data Collection in Multi-UAV-Assisted Wireless Sensor Networks," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*. Harbin City, China: IEEE, Jun. 2021, pp. 669–674.

[23] " Coordinated multi-point operation for LTE physical layer aspects," 3GPP, 3GPP TR 36.819, Sep. 2013.

[24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Aug. 2017.

[25] J. Baxter and P. L. Bartlett, "Infinite-Horizon Policy-Gradient Estimation," *Journal of Artificial Intelligence Research*, vol. 15, pp. 319–350, Nov. 2001.