Efficient Communications in Multi-Agent Reinforcement Learning for Mobile Applications

Zefang Lv, *Student Member, IEEE,* Liang Xiao, *Senior Member, IEEE,* Yousong Du, Yunjun Zhu, Shuai Han, *Senior Member, IEEE,* and Yong-Jin Liu, *Senior Member, IEEE*

***Abstract*—The environment observations and learning experiences shared by the cooperative learning agents accelerate multi-agent reinforcement learning (MARL) with partial observations for mobile applications but the performance degrades due to the redundant and outdated observations under severe channel fading in wireless networks. In this paper, we propose an efficient communication scheme in MARL for mobile applications that enables each learning agent to optimize the cooperative agents and the learning parameters to integrate the shared information. The cooperative agents are chosen according to the learning environment observations, the channel states, and the task similarity with neighboring agents. The learning parameters are chosen based on the attention mechanism that exploits the correlation with the local observation to enhance the agent receptive field for efficient policy exploration. Neural networks with weights updated based on the learning factors determined by the task similarity are designed to further improve the learning efficiency. The performance bounds including the information gain from the learning agent cooperation, the communication cost and the utility are provided based on the Nash equilibrium of the cooperative MARL communication game. The proposed scheme is implemented in the anti-jamming video transmission of the unmanned aerial vehicle swarms to optimize the transmit channel and power and experimental results verify the performance gain over the benchmark.**

***Index Terms*—Multi-agent reinforcement learning, communications, unmanned aerial vehicle, mobile applications, wireless networks.**

I. Introduction

THE latest development of multi-agent reinforcement learning (RL) has enabled mobile devices as the learning agents to optimize their decisions such as the radio resource allocation and trajectory design to support a variety of mobile applications and services in wireless networks [1], [2]. However, most learning agents only have partial observations on the environment such as the radio channel states and suffer from non-stationarity caused by the changing policies of other agents, which in turn degrades the RL performance and the quality of service (QoS) for partially observable Markov decision process (POMDP) in dynamic networks [3].

As the neighboring learning agents share a common environment and can exchange their environment observations and learning experiences via the wireless networks, the multi-agent RL (MARL) algorithms can exploit the learning information shared from the cooperative agents in the state formulation and learning parameter update to enhance the agent receptive field and improve the RL policy optimization efficiency [4], [5]. For example, the multi-agent deep Q-network (DQN)-based interference mitigation scheme in [5] exploits the shared observations such as the signal-to-interference-plus-noise ratio (SINR) and the weights of associated user equipment to optimize the downlink transmit power and user scheduling and thus improves the data rate in wireless networks against interference from neighboring access points.

Both the multi-agent communication efficiency and RL performance depend on the cooperative agent selection, which in turn relies on the accurate network states that are rarely known by most mobile devices. As a seminal work on multi-agent communications, I2C proposed in [6] applies a prior network consisting of a feed-forward neural network to evaluate the belief of the neighboring agents based on local observations, such as the positions of the neighboring agents in the navigation task, to decide which agent to communicate with. However, the performance of I2C is hindered by long communication delay and observation redundancy due to the limited bandwidth and time-varying channel states, leading to a degradation in RL performance and application QoS in large-scale wireless networks.

In this paper, we propose an efficient communication scheme in MARL for mobile applications that optimizes the cooperative agents to share environment observations and learning parameters to integrate the shared learning information to accelerate task learning with restricted communication overhead. The cooperative agents are chosen based on a communication state that consists of the local environment observation of the learning agent, the radio channel states, the task similarity with neighboring agents and previous learning performance to maximize the expected long-term discounted communication reward as the weighted sum of RL task reward and communication cost.

A scaled dot-product attention-based observation encoder compares the shared and local observations to dynamically extract observation correlations and enhance the receptive field for learning agents, while reducing the redundancy in the RL task state formulation to accelerate the task policy selection.

This work was supported in part by the Natural Science Foundation of China under Grant U21A20444 and the National Key Research and Development Program of China under Grant 2023YFB3107603, and in part by the Natural Science Foundation of China under Grants 62332019 and U2336214. (*Corresponding author: Liang Xiao*)

Zefang Lv, Liang Xiao, Yousong Du and Yunjun Zhu are with the Department of Informatics and Communication Engineering & Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Xiamen University, Xiamen 361005, China. Email: lxiao@xmu.edu.cn

Shuai Han is with the School of Electronics and Information Engineering, Harbin Institute of Technology, China. Email: hanshuai@hit.edu.cn

Yong-Jin Liu is with the Department of Computer Science and Technology, Tsinghua University. Email: liuyongjin@tsinghua.edu.cn

The local task learning parameters are updated based on both the local task reward and the shared learning parameters with learning factors determined by the task similarity to further improve the task policy exploration efficiency.

A communication scheme for deep MARL that exploits both the observations and learning parameters shared from the selected cooperative agent is proposed to accelerate learning of neural networks in the task policy optimization. A sequential architecture with fully-connected layer based neural networks in the communication learning layer estimates the expected long-term communication reward to choose the cooperative agents for network weights and observation sharing. The neural network for task policy optimization with weights updated based on the shared learning experiences and their corresponding importance adaptively determined by the task similarity are designed to further improve the learning efficiency.

We formulate a cooperative MARL communication game among learning agents based on the informational model presented in [7] to provide the communication learning performance bounds. In this game, each learning agent chooses whether to communicate with neighboring agents with the goal of maximizing the RL task performance with less communication cost. Based on the Nash equilibrium of the game, the performance bounds including the information gain, the communication cost and the utility are provided and verified via simulations, in which the information gain obtained from the learning agent cooperation increases with the task similarity with neighboring agents and shared information of the selected cooperative agents.

As a case study, we implement the proposed communication schemes in MARL-based anti-jamming video transmission of unmanned aerial vehicle (UAV) swarm to optimize the transmit channel and power. Simulations are performed in a 10-UAV swarm based on the collected channel states of the swarm communication and jammer-UAV links to show the performance gain of the proposed scheme over the benchmark I2C in [6]. In addition, the proposed scheme is also implemented in a 5-UAV swarm equipped with Raspberry Pi 4 against a jammer, i.e., a universal software radio peripheral (USRP) N210 controlled by a laptop. Both the simulation and experimental results show that the proposed communication scheme improves the MARL-based UAV swarm anti-jamming video transmission performance with reduced packet loss rate (PLR), transmission delay, UAV transmit power and convergence time over the benchmark I2C in [6].

The main contributions of this paper are summarized as follows:

- We propose an efficient communication scheme to improve both the learning and communication efficiency for MARL-based mobile applications, which enables each learning agent to optimize the cooperative agents for environment observation and learning experience sharing and the parameters to integrate the shared information in the RL task policy learning.
- We formulate the interactions among learning agents as a cooperative MARL communication game based on the informational model and provide the performance bounds including the information gain from the learning information sharing, the communication cost and utility based on the Nash equilibrium of the game.
- We implement the MARL with the proposed scheme in the UAV swarm anti-jamming video transmission system to optimize the transmit channel and power. Both simulations and experiments are performed to show the performance gain in terms of PLR and transmission delay over the benchmark.

The rest of this paper is organized as follows. We review the related work in Section II, and present the system model in Section III. An efficient communication scheme for MARL and a version for deep MARL are presented in Sections IV and V. The performance analysis and a case study is given in Sections VI and VII, respectively. Simulation and experimental results are reported in Sections VIII and IX, followed by the conclusion in Section X.

II. Related Work

MARL such as multi-agent deep deterministic policy gradient (DDPG) enables multi-agent systems in wireless networks to optimize transmission policies such as resource allocation and trajectory design [1], [2], [8]–[11]. For example, the multi-UAV assisted mobile edge computing system in [2] applies the multi-agent twin delayed DDPG to optimize the trajectory design, task allocation, and power management and thus decreases the total system cost. The full-duplex multi-UAV network system in [8] uses a clip and count-based proximal policy optimization algorithm to optimize the decoupled uplink-downlink association and trajectory design in a distributed manner. However, fully decentralized learning or centralized training and decentralized executing have learning performance degradation under partial observation in large-scale wireless networks.

Learning information exchange including environment observations and learning experiences among learning agents such as mobile devices enhances the RL performance under partial observation [5], [7], [12]–[18]. For example, the MARL-based resource management scheme in [5] enables the radio devices to exchange observations with their neighboring agents to improve the learning performance of power control and user association. The multi-agent wireless system in [7] shares information with other agents via neural network parameters and analyzes the effect of the parameter sharing frequency on the convergence speed. A multi-agent DQN based cellular offloading scheme in [16] optimizes trajectory design and power allocation with experiences shared via a neural network and decreases the convergence time. However, the full communication protocols above that share observations or learning parameters among all the learning agents may lead to higher communication overhead and introduce information redundancy in the task learning process, thus degrading the learning performance and speed.

Efficient communication is promising for MARL to enhance learning performance with attention experience sharing, and as examples, RIAL and DIAL in [3] and CommNet in [19] learn to communicate between agents in cooperative environments

with partial observations. For example, the communication protocol in [3] uses deep Q-learning with recurrent networks to encode past observations, actions and local observations into binary messages and thus reduces communication overhead but has information loss in large-scale wireless networks. The communication information mainly consists of original observations, encoded observations [4], [20] and intended actions [21]. In particular, an intention sharing scheme in [21] applies the attention mechanism to generate an imagined trajectory based on the shared messages from other agents and enhance the coordination of multi-agent systems.

Learning cooperative agent selection and integration of shared information further enhance the learning efficiency [6], [22]. For example, an attentional communication scheme in [22] uses a bidirectional long short-term memory unit to learn when to communicate and how to integrate the received information for cooperative decision making. An individually inferred communication scheme in [6] applies causal inference to pretrain a prior network for agent-agent communications, which maps the local observation to a belief value to determine which agent to communicate with. However, the limited radio bandwidth and time-varying channel states resulting in long communication delay may degrade the RL performance and application QoS in large-scale wireless networks.

Existing communication schemes in MARL also consider realistic constraints such as communication cost and dynamic environments [23]–[27]. For example, an information bottleneck based communication scheme in [24] jointly optimizes which agent is communicating what message and to whom under limited bandwidth and estimates the posterior distribution of messages based on the variational information bottleneck. An efficient communication scheme in [25] investigates communications among multiple agents over a noisy channel and jointly optimizes the channel coding, source coding and joint source-channel coding problems to enhance coordination and cooperation among agents. However, these communication schemes based on fully connected structure suffer from higher communication overhead and QoS degradation for mobile applications in large-scale wireless networks.

Compared with MARL communication schemes mentioned above, we optimize the cooperative agents for learning information sharing based on task similarity and radio channel states with neighboring agents and the information integration method to enhance both the learning and communication efficiency for mobile applications in large-scale dynamic wireless networks. In our previous work in [28], an efficient communication scheme for MARL is proposed, which enables each learning agent to optimize the cooperative agent selection and the task state formulation to improve learning performance and QoS of RL-based applications in wireless networks. In this paper, we further propose a sequential architecture to choose cooperative agents with lower computational complexity and design neural networks with weight updated with learning factor determined by the task similarity to improve the learning efficiency. The proposed schemes are implemented in the UAV swarm anti-jamming video transmission to optimize transmit channel and power and experimental results verify the performance gain over the benchmark.

TABLE I: Summary of Symbols and Notations.

Symbol	Description
N	Number of neighboring agents
$\boldsymbol{o}^{(k)}$	Local observation of the learning agent at time slot k
$\hat{\boldsymbol{o}}_i^{(k)}$	Received observation from agent i
$\tilde{\boldsymbol{o}}_i^{(k)}$	Encoded observation of agent i
$\bar{\boldsymbol{o}}^{(k)}$	Previous received observations
$\eta_i^{(k)}$	RL task similarity with agent i
$h_i^{(k)}$	Channel gain with agent i
$\tilde{\boldsymbol{s}}^{(k)}$	Communication learning state
$\boldsymbol{x}^{(k)}$	Selected cooperative agents
$\boldsymbol{s}^{(k)}$	Task learning state
$\boldsymbol{a}^{(k)}$	Task action
$\boldsymbol{\rho}^{(k)}$	Task learning performance
$r^{(k)}$	Task reward
$\xi^{(k)}$	Number of selected cooperative agents
$\tilde{r}^{(k)}$	Communication reward
$p^{(k)}$	Transmit power of the UAV
$f^{(k)}$	Transmit channel of the UAV
$b^{(k)}$	UAV message size
$\varsigma^{(k)}$	Jamming power received at the control center
$\rho_1^{(k)}$	SINR of the UAV signals received by the control center
$\rho_2^{(k)}$	Transmission delay
$\rho_3^{(k)}$	Packet loss rate
$\varrho^{(k)}$	Frequency hopping cost

III. System Model

A. MARL Model for Mobile Applications

In a POMDP, N learning agents communicate via wireless channels to cooperatively perform learning tasks of mobile applications such as target tracking and collaborative sensing. At time slot k, learning agent i obtains local observation $\boldsymbol{o}_i^{(k)}$, selects an action $\boldsymbol{a}_i^{(k)} \in \mathcal{A}_i$ that follows the policy distribution $\pi_i \in [0,1]^{|\mathcal{A}_i|}$, receives a reward $r_i^{(k)}$ from the environment and transitions to a new state $\boldsymbol{s}_i^{(k+1)}$. The goal is to learn the policy π_i that maximizes the expected long-term discounted reward $R_i^{(k)} = \sum_{n=k}^{\infty} \gamma^{n-k} r_i^{(n)}$, where γ is the discount factor that represents the importance of the immediate reward and the future reward. The POMDP is modeled by a tuple $\mathcal{G} = \{\mathcal{S}, \mathcal{O}_i, \mathcal{A}_i, \mathcal{P}, r_i, \pi_i, \}_{1 \le i \le N}$, where

- State space $\mathcal{S}$ represents all possible state configurations of the environment and learning agents.
- Observation set $\mathcal{O}_i$ consists of all local observations of learning agent i.
- Action set $\mathcal{A}_i$ consists of all feasible task policies of learning agent i.
- $\mathcal{P}$: $\mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \times \mathcal{S} \rightarrow [0,1]$ describes the distribution of possible next state given the current state and actions of all learning agents.
- Reward r_i: $\mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \rightarrow \mathbb{R}$ is the immediate reward as the weighted sum of task performance such as

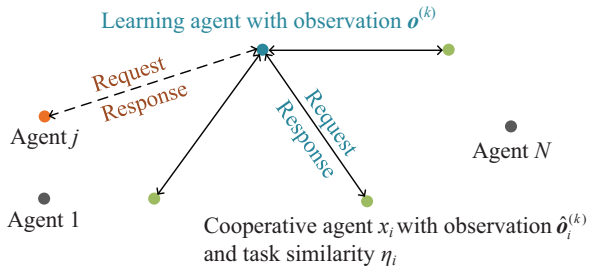


Fig. 1. A wireless network with $N+1$ cooperative agents performing RL-based tasks such as target tracking, in which the learning agent selects cooperative agents $\boldsymbol{x}^{(k)}$ to request learning information such as environment observations and learning experiences from neighboring agents to enhance the performance of mobile applications.

video transmission quality and target tracking accuracy, which depends on the global state and actions of all agents.

- Policy distribution π_i: $\mathcal{S} \times \mathcal{A}_i \rightarrow [0,1]^{|\mathcal{A}_i|}$ represents the probability of learning agent i to choose the action $\boldsymbol{a}_i \in \mathcal{A}_i$ in state $\boldsymbol{s}_i$.

B. Network and Communication Model for Learning Agents

Without loss of generality, each learning agent is assumed to have N neighboring agents $\mathcal{N} = \{1, 2 \ldots, N\}$ and exchanges learning messages including observations and learning parameters with neighboring agents within the communication range based on the carrier sense multiple access with collision avoidance access via the wireless channels. As shown in Fig. 1, at time slot k, the learning agent observes local observation $\boldsymbol{o}^{(k)} \in \mathbb{R}^O$, estimates the channel states $\boldsymbol{h}^{(k)} = [h_{1\leq i\leq N}^{(k)}]$ and evaluates the task similarity with neighboring agents $\boldsymbol{\eta}^{(k)} = [\eta_{1\leq i\leq N}^{(k)}] \in [0,1]^N$ based on the Jensen-Shannon distance between local and shared policy distribution or Bisimulation relation of previous shared observations as [29], [30]. For example, an encoder with convolutional layers learns the task-relevant information from local observations $\boldsymbol{o}^{(k)}$ and previous shared observations $\bar{\boldsymbol{o}}_i$ to evaluate the similarity as a Bisimulation metric $d(\boldsymbol{o}^{(k)}, \bar{\boldsymbol{o}}_i)$ as presented in [31].

The learning agent chooses at most M cooperative agents $\boldsymbol{x}^{(k)} \subset \mathcal{N}$ to request the learning information and formulate the task learning state for more efficient RL task policy optimization. Specifically, the learning agent collects information about neighboring agents when participating in a new RL task, such as the device identities, learning state configuration and task policy from the previous learning response messages of other agents. After determining the cooperative agents $\boldsymbol{x}^{(k)}$ from N neighboring agents $\mathcal{N} = \{1, 2 \ldots, N\}$, the selected agents ID and required information such as the observation and current learning parameters, e.g., the task Q-values or policy distribution, are broadcasted to the selected agents $\boldsymbol{x}^{(k)}$ in the learning request message.

Upon receiving the learning request message from the learning agent, the cooperative agent $i \in \boldsymbol{x}^{(k)}$ obtains its local observation $\hat{\boldsymbol{o}}_i^{(k)}$ and current learning parameters and sends the learning information in the learning response message back to the learning agent. If neighboring agent i receives multiple cooperative requests from other agents, the current learning information will be sent to these request learning agents in the leaning response message sequentially.

The cooperative states $\left\{\hat{\boldsymbol{o}}_i^{(k)} | i \in \boldsymbol{x}^{(k)}\right\}$ are exploited to formulate the task learning state $\boldsymbol{s}^{(k)}$, which is used as the basis for the learning agent to choose the task action $\boldsymbol{a}^{(k)} \in \mathcal{A}$. Based on the task performance $\boldsymbol{\rho}^{(k)}$ such as the PLR of video transmission and message bit error rate (BER), the learning agent evaluates the task reward $r^{(k)}$ and communication cost, i.e., the number of selected cooperative agents denoted by $\xi^{(k)}$, to guide the cooperative agent selection. Time slot k and agent ID i are omitted if no confusion occurs in the subsequent sections because our proposed algorithms are applicable to each learning agent and only depend on the latest learning information such as observations, learning performance and parameters instead of all the learning information of previous time slots.

IV. Efficient Communications for MARL

We propose a learning based efficient multi-agent communication scheme with sequential architecture named ECOM for MARL-based mobile applications in wireless networks that enables learning agents to choose their cooperative agents $\boldsymbol{x}^{(k)}$ and task action $\boldsymbol{a}^{(k)}$ for faster task learning under partial observation. The cooperative agents $\boldsymbol{x}^{(k)}$ are chosen in each time slot k based on a communication state $\tilde{\boldsymbol{s}}^{(k)}$ including local observation $\boldsymbol{o}$, channel states $\boldsymbol{h}$ and task similarity $\boldsymbol{\eta}$ with neighboring agents, and previous communication cost ξ. According to the attention mechanism similar to [32], the correlation between the local and shared observations denoted by $\tilde{\boldsymbol{o}}_i$, $i \in \boldsymbol{x}^{(k)}$, is calculated based on the attention function, i.e., $\tilde{\boldsymbol{o}}_i = m(\boldsymbol{o}, \boldsymbol{o}_i)$, to reduce the information redundancy in the formulation of the task learning state $\boldsymbol{s}^{(k)}$ for task policy selection. In addition, the shared learning parameters such as task Q-values are exploited to update the local task Q-values based on the learning factors determined by the task similarity for efficient task policy exploration.

As shown in Fig. 2, the MARL communication scheme ECOM jointly chooses the cooperative agents $\boldsymbol{x}^{(k)}$ and task action $\boldsymbol{a}^{(k)}$ to maximize the discounted long-term reward of the learning agent. The optimal cooperative agent selection that determines both the multi-agent communication efficiency and RL performance depends on the accurate network states and perfect knowledge on the learning information of neighboring agents, which are rarely known by most mobile devices accurately. Therefore, the communication learning layer adaptively chooses the cooperative agents $\boldsymbol{x}^{(k)}$ based on the communication state $\tilde{\boldsymbol{s}}^{(k)}$ and communication policy distribution π^{C} for mobile devices under time-varying channel states via ϵ-greedy based learning method. Specifically, the communication state $\tilde{\boldsymbol{s}}^{(k)}$ consists of local observations $\boldsymbol{o}$, channel states $\boldsymbol{h}$, task similarity $\boldsymbol{\eta}$, previous observations of the W neighboring agents $\bar{\boldsymbol{o}}$ and previous communication cost ξ, i.e.,

$$\tilde{\boldsymbol{s}}^{(k)} = [\boldsymbol{o}, \boldsymbol{h}, \boldsymbol{\eta}, \bar{\boldsymbol{o}}, \xi] \in \hat{\mathbf{S}} \tag{1}$$

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.2024.3392608

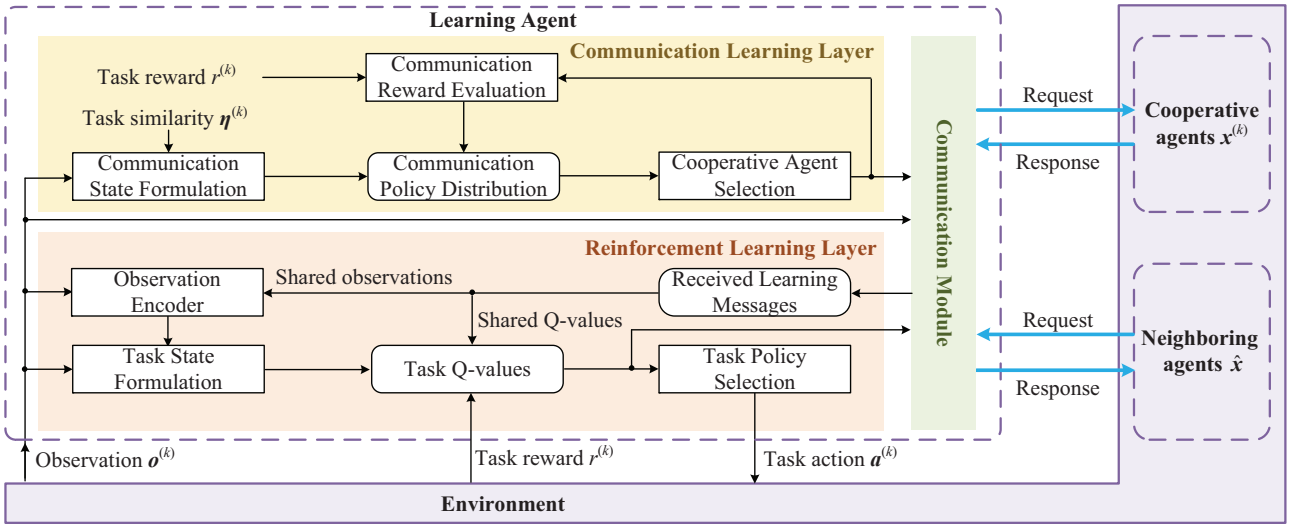


Fig. 2. Illustration of the MARL communication scheme, in which the learning agent chooses the cooperative agents $\boldsymbol{x}^{(k)}$ in the communication learning layer, request and send learning information to neighboring agents via the communication module and selects the task action $\boldsymbol{a}^{(k)}$ in the reinforcement learning layer based on the aggregated task learning state $\boldsymbol{s}^{(k)}$.

Algorithm 1 Efficient communications for MARL

Initialize $\bar{\boldsymbol{o}}$, ξ, λ, γ, ϖ and $\boldsymbol{\omega}$, $\boldsymbol{Q}^{\mathrm{R}}$ and $\boldsymbol{Q}^{\mathrm{C}}_{1\le j\le M}$
for $k = 1, ..., K$ **do**
  Obtain local observation $\boldsymbol{o}^{(k)}$
  Estimate channel gain $\boldsymbol{h}^{(k)}$ and task similarity $\boldsymbol{\eta}^{(k)}$ with neighboring agents
  Formulate communication state $\tilde{\boldsymbol{s}}^{(k)}$ via (1)
  for $j = 1, ..., M$ **do**
    Select $x_j^{(k)}$ via (2)
  end for
  Send learning request to agents $\boldsymbol{x}^{(k)}$ and receive $\left\{\left\{\hat{\boldsymbol{o}}_i, \hat{\boldsymbol{Q}}_i^{\mathrm{R}}\right\} | i \in \boldsymbol{x}^{(k)}\right\}$
  Calculate $\tilde{\boldsymbol{o}}_i = m(\boldsymbol{o}, \hat{\boldsymbol{o}}_i)$, $\forall i \in \boldsymbol{x}^{(k)}$
  Formulate task state $\boldsymbol{s}^{(k)} = \left[\boldsymbol{o}^{(k)}, [\tilde{\boldsymbol{o}}_i]_{i\in\boldsymbol{x}^{(k)}}\right]$
  Select $\boldsymbol{a}^{(k)}$ via (4)
  Receive task reward $r^{(k)}$
  Update $Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}\right)$ via (5)
  Update $\boldsymbol{Q}^{\mathrm{R}}$ via (6)
  Calculate $\tilde{r}^{(k)}$ via (7)
  for $j = 1, ..., M$ **do**
    Update $Q_j^{\mathrm{C}}\left(\tilde{\boldsymbol{s}}^{(k)}, x_j^{(k)}\right)$ similar to (5)
  end for
end for

A sequential cooperative agent selection architecture is designed for large-scale networks to accelerate the communication policy learning for choosing a large number of cooperative agents. This architecture decomposes the communication policy $\boldsymbol{x}^{(k)}$ into M levels to sequentially choose at most M cooperative agents from neighboring agents $\mathcal{N} = \{1, 2 \ldots, N\}$, which significantly decreases the computational complexity of the communication policy selection and learning. Specifically, level j consists of an $|\tilde{\mathbf{S}}| \times (N+1)$ Q-table to output the Q-values of the j-th cooperative agent under current state $\tilde{\boldsymbol{s}}^{(k)}$ denoted by $Q_j^{\mathrm{C}}\left(\tilde{\boldsymbol{s}}^{(k)}, \cdot\right)$ to choose $x_j \in \{0, 1, 2, \cdots, N\}$, where $x_j = n \in \mathcal{N}$ indicates that neighboring agent n is selected in level j and $x_j = 0$ indicates no selected agent in this level. Based on the ϵ-greedy method, the communication policy distribution in level j is designed to make a trade-off between exploration and exploitation in the communication policy learning, which is formulated as

$$\Pr\left(x_j = \hat{x}\right) = \begin{cases} 1-\epsilon, & \hat{x} = \underset{x\in\{0,1,\cdots,N\}}{\operatorname{argmax}} Q_j^{\mathrm{C}}\left(\tilde{\boldsymbol{s}}^{(k)}, x\right) \\ \frac{1}{N}, & \hat{x} \neq \underset{x\in\{0,1,\cdots,N\}}{\operatorname{argmax}} Q_j^{\mathrm{C}}\left(\tilde{\boldsymbol{s}}^{(k)}, x\right) \end{cases} \tag{2}$$

Upon receiving the learning response message from the selected cooperative agents $\boldsymbol{x}^{(k)}$, the shared observation $\hat{\boldsymbol{o}}_i$ and task Q-values $\boldsymbol{Q}_i^{\mathrm{R}}$, $i \in \boldsymbol{x}^{(k)}$, are extracted to assist task learning. The observation encoder $m(\boldsymbol{o}, \hat{\boldsymbol{o}}_i)$ applies the scaled dot-product attention mechanism in [32] to compare the local observation $\boldsymbol{o}$ with the shared observation $\hat{\boldsymbol{o}}_i$, $i \in \boldsymbol{x}^{(k)}$, and extract the importance of the shared observation, which is used to encode observation $\hat{\boldsymbol{o}}_i$ in the formulation of the task learning state $\boldsymbol{s}^{(k)}$. Specifically, local observation $\boldsymbol{o}$ and shared observations $\hat{\boldsymbol{o}}_i$ from cooperative agent i both serve as the input of the observation encoder. The local observation $\boldsymbol{o}$ is transformed to *query* with $\boldsymbol{W}^{\mathrm{Q}} \in \mathbb{R}^{O\times V}$ and the shared observations $\hat{\boldsymbol{o}}_i$ is transformed to *key* and *value* with $\boldsymbol{W}^{\mathrm{K}}$ and $\boldsymbol{W}^{\mathrm{V}} \in \mathbb{R}^{O\times V}$, respectively. The encoded observation denoted by $\tilde{\boldsymbol{o}}_i$ depends on the weighted sum of *value* and the dot product and softmax function is applied to calculate the weight for cooperative agent $i \in \boldsymbol{x}^{(k)}$ denoted by $\boldsymbol{\alpha}_i = [\alpha_{i,v}]_{v=1}^{V}$, i.e.,

$$\alpha_{i,v} = \frac{\exp\left(\boldsymbol{o}\boldsymbol{W}_v^{\mathrm{Q}}\hat{\boldsymbol{o}}_i\boldsymbol{W}_v^{\mathrm{K}}\right)}{\sum_{v=1}^{V}\exp\left(\boldsymbol{o}\boldsymbol{W}_v^{\mathrm{Q}}\hat{\boldsymbol{o}}_i\boldsymbol{W}_v^{\mathrm{K}}\right)} \tag{3}$$

Based on the shared observation $\hat{\boldsymbol{o}}_i$ and its transforming matrix $\boldsymbol{W}^{\mathrm{V}}$ and weights $\boldsymbol{\alpha}_i$, the encoded observation $\tilde{\boldsymbol{o}}_i = \hat{\boldsymbol{o}}_i\boldsymbol{W}^{\mathrm{V}}\boldsymbol{\alpha}_i^{\mathrm{T}}$.

The reinforcement learning layer optimizes the specific RL task action $\boldsymbol{a}^{(k)}$ based on the task learning state $\boldsymbol{s}^{(k)}$, task Q-values $Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k)},\cdot\right)$ and task policy distribution π^{R}. Specifically, the task state $\boldsymbol{s}^{(k)}$ consists of local observation $\boldsymbol{o}$ and encoded shared observations of the cooperative agents, i.e., $\boldsymbol{s}^{(k)}=[\boldsymbol{o},[\tilde{\boldsymbol{o}}_i]_{i\in\boldsymbol{x}^{(k)}}]$. The task policy $\boldsymbol{a}\in\mathcal{A}$ is chosen based on the ϵ-greedy method according to the Q-values of task state-action pairs $Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k)},\cdot\right)$ or the task policy distribution π^{R} with

$$\Pr\left(\boldsymbol{a}^{(k)}=\hat{\boldsymbol{a}}\right)=\pi^{\mathrm{R}}\left(\hat{\boldsymbol{a}}\left|\boldsymbol{s}^{(k)}\right.\right) \tag{4}$$

Upon receiving the task feedback from the environment, task reward $r^{(k)}$ is evaluated based on the task performance $\boldsymbol{\rho}$ such as the video transmission quality. The Q-value in the task layer $Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k)},\boldsymbol{a}^{(k)}\right)$ is updated with reward $r^{(k)}$ by Bellman equation

$$\begin{aligned}Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k)},\boldsymbol{a}^{(k)}\right)\leftarrow&\lambda\left(r^{(k)}+\gamma\max_{\hat{\boldsymbol{a}}\in\mathcal{A}}Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k+1)},\hat{\boldsymbol{a}}\right)\right)\\&+(1-\lambda)Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k)},\boldsymbol{a}^{(k)}\right),\end{aligned} \tag{5}$$

where λ and γ represents the importance of the received reward and the future reward in the learning process, respectively. In addition, the shared learning information such as Q-values $\hat{\boldsymbol{Q}}_i^{\mathrm{R}}$ are also exploited to update local Q-values, i.e.,

$$\boldsymbol{Q}^{\mathrm{R}}\leftarrow\frac{\boldsymbol{Q}^{\mathrm{R}}+\sum_{i\in\boldsymbol{x}^{(k)}}\omega_i\hat{\boldsymbol{Q}}_i^{\mathrm{R}}}{1+\sum_{i\in\boldsymbol{x}^{(k)}}\omega_i} \tag{6}$$

The learning factor ω_i for cooperative agent i is adaptively determined based on the task similarity or previous communication performance to further improve the task learning efficiency. For example, the learning factor ranging from 0 to 1 increases linearly with the task similarity between the learning agent and cooperative agent i.

The communication cost, i.e., the number of the selected cooperative agents $\xi^{(k)}$, is evaluated to calculate the communication reward $\tilde{r}^{(k)}$, which increases with task reward $r^{(k)}$ and decreases with the communication cost $\xi^{(k)}$ and the corresponding importance factor ϖ^{C}, i.e.,

$$\tilde{r}^{(k)}=r^{(k)}-\varpi^{\mathrm{C}}\xi^{(k)} \tag{7}$$

The Q-values in the M levels of the communication learning layer $Q_j^{\mathrm{C}}\left(\tilde{\boldsymbol{s}}^{(k)},x_j^{(k)}\right)$, $1\le j\le M$ are updated with communication reward $\tilde{r}^{(k)}$ similar to (5).

V. Efficient Communications for Deep MARL

We further propose an efficient communication scheme with sequential architecture for the deep MARL that exploits the shared learning parameters to further accelerate the communication and task policy learning. The communication Q-values of each level in the cooperative agent selection are estimated via the fully connected neural networks to compress the high-dimension state for the large-scale wireless networks. The shared learning parameters such as the neural network weights are exploited to update the task network for fast learning.

The communication learning layer chooses the cooperative agent $\boldsymbol{x}^{(k)}$ via the sequential cooperative agent selection architecture consisting of M levels. Each level j consists of a neural network named C-Network j with weight $\boldsymbol{\alpha}_j$ including four fully-connected layers, i.e., an input layer with $(W+1)O+2N+1$ nodes, two hidden layers with $g_{j,1}$ and $g_{j,2}$ nodes and an output layer with $N+1$ nodes. With the communication state $\tilde{\boldsymbol{s}}^{(k)}$ similar to (1) as input, the C-Network j estimates the Q-values $Q_j^{\mathrm{C}}\left(\tilde{\boldsymbol{s}}^{(k)},\cdot;\boldsymbol{\alpha}_j\right)$ to calculate the communication policy distribution via (2).

Algorithm 2 Efficient communications for deep MARL

Initialize $\bar{\boldsymbol{o}}$, ξ, λ, γ, ϖ, $\boldsymbol{\theta}$, $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}_{1\le j\le M}$
for $k=1,...,K$ **do**
  Same as lines 3-5 in Algorithm 1
  for $j=1,...,M$ **do**
    Input $\tilde{\boldsymbol{s}}^{(k)}$ to C-Network j and obtain $Q_j^{\mathrm{C}}\left(\tilde{\boldsymbol{s}}^{(k)},\cdot;\boldsymbol{\alpha}_j\right)$
    Select $x_j^{(k)}$ via (2)
  end for
  Send learning request to agents $\boldsymbol{x}^{(k)}$ and receive $\left\{\left\{\hat{\boldsymbol{o}}_i,\hat{\boldsymbol{\theta}}_i\right\}|i\in\boldsymbol{x}^{(k)}\right\}$
  Formulate $\boldsymbol{s}^{(k)}$ similar to Algorithm 1
  Input $\boldsymbol{s}^{(k)}$ to task network and obtain $Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k)},\cdot\right)$
  Select task action $\boldsymbol{a}^{(k)}$ via (4)
  Receive task reward $r^{(k)}$
  $\mathcal{D}^{\mathrm{R}}\leftarrow\mathcal{D}^{\mathrm{R}}\bigcup\boldsymbol{z}^{\mathrm{R}}$
  if $|\mathcal{D}^{\mathrm{R}}|\ge G$ **then**
    Randomly sample G experiences from $\mathcal{D}^{\mathrm{R}}$ to update $\boldsymbol{\theta}$ via (9)
  end if
  Update $\boldsymbol{\theta}$ via (10)
  Evaluate communication reward $\tilde{r}^{(k)}$ via (7)
  $\mathcal{D}^{\mathrm{C}}\leftarrow\mathcal{D}^{\mathrm{C}}\bigcup\boldsymbol{z}^{\mathrm{C}}$
  if $|\mathcal{D}^{\mathrm{C}}|\ge G$ **then**
    Randomly sample G experiences from $\mathcal{D}^{\mathrm{C}}$
    for $j=1,...,M$ **do**
      Update $\boldsymbol{\alpha}_j$ similar to (9)
    end for
  end if
end for

The task learning state $\boldsymbol{s}^{(k)}$ is formulated as in Algorithm 1 and input to the task network such as a fully-connected neural network to obtain the Q-values of state-action pairs $Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k)},\cdot\right)$. The task action $\boldsymbol{a}\in\mathcal{A}$ is chosen according to the task policy distribution $\pi^{\mathrm{R}}\left(\cdot|\boldsymbol{s}^{(k)};\boldsymbol{\theta}\right)$ formulated based on $Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k)},\boldsymbol{a}\right)$ with ϵ-greedy method. Upon receiving the task feedback, task reward $r^{(k)}$ is evaluated to update the network weights $\boldsymbol{\theta}$ with the experience replay technique. The task learning experience including the task state $\boldsymbol{s}^{(k)}$, task action $\boldsymbol{a}^{(k)}$, task reward $r^{(k)}$ and next task state $\boldsymbol{s}^{(k+1)}$, i.e.,

$$\boldsymbol{z}^{\mathrm{R}}=\left\{\boldsymbol{s}^{(k)},\boldsymbol{a}^{(k)},r^{(k)},\boldsymbol{s}^{(k+1)}\right\} \tag{8}$$

The task memory pool $\mathcal{D}^{\mathrm{R}}$ stores the experience $\boldsymbol{z}^{\mathrm{R}}$ and G experiences are randomly and uniformly sampled from $\mathcal{D}^{\mathrm{R}}$ to formulate the minibatch $\mathcal{B}^{\mathrm{R}}$. The Q-values $Q^{\mathrm{R}}\left(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta}\right)$ in the reinforcement learning layer estimated by the task network are used to update the network weights $\boldsymbol{\theta}$ via stochastic gradient descent algorithm such as Adam optimizer with learning rate

β by minimizing the loss function given by

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\{\boldsymbol{s},\boldsymbol{a},r,\bar{\boldsymbol{s}}\}\in\mathcal{B}^{\mathrm{R}}}\Bigg[\Bigg(r + \gamma \max_{\boldsymbol{a}\in\mathcal{A}} Q^{\mathrm{R}}\Big(\bar{\boldsymbol{s}}, \boldsymbol{a}; \boldsymbol{\theta}\Big) - Q^{\mathrm{R}}\Big(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta}\Big)\Bigg)^2\Bigg] \tag{9}$$

The shared network weights $\{\hat{\boldsymbol{\theta}}_i | i \in \boldsymbol{x}^{(k)}\}$ are also applied to update the local task network weights, i.e.,

$$\boldsymbol{\theta} \leftarrow \frac{\boldsymbol{\theta} + \sum_{i\in\boldsymbol{x}^{(k)}} \omega_i \hat{\boldsymbol{\theta}}_i}{1 + \sum_{i\in\boldsymbol{x}^{(k)}} \omega_i} \tag{10}$$

Upon receiving the task reward $r^{(k)}$, the communication reward $\tilde{r}^{(k)}$ is evaluated based on the task reward $r^{(k)}$ and the number of the selected cooperative agents $\xi^{(k)}$. The communication experience is formulated based on the communication state sequence $\tilde{\boldsymbol{s}}^{(k)}$, communication policy $\boldsymbol{x}^{(k)}$, communication reward $\tilde{r}^{(k)}$ and next communication state $\tilde{\boldsymbol{s}}^{(k+1)}$, i.e.,

$$\boldsymbol{z}^{\mathrm{C}} = \left\{\tilde{\boldsymbol{s}}^{(k)}, \boldsymbol{x}^{(k)}, \tilde{r}^{(k)}, \tilde{\boldsymbol{s}}^{(k+1)}\right\}, \tag{11}$$

which is stored in the memory pool $\mathcal{D}^{\mathrm{C}}$ to update network weights $\boldsymbol{\alpha}_{1\le j\le M}$ in the communication layer. By sampling G experiences from $\mathcal{D}^{\mathrm{C}}$ to formulate the minibatch $\mathcal{B}^{\mathrm{C}}$, the neural network weight $\boldsymbol{\alpha}_j$ of level j is updated by minimizing the loss function similar to (9).

VI. Performance Analysis

We formulate the interactions among the learning agents as a cooperative MARL communication game and analyze the performance bounds including the information gain from the learning agent cooperation, the communication cost and utility based on the Nash equilibrium of the game.

A. Game Model

In the cooperative MARL communication game, N learning agents $\mathcal{N} = \{1, 2, \cdots, N\}$ as players communicate with each other to share learning information. Specifically, learning agent i chooses whether to communicate with neighboring agents $\mathcal{N}_{-i}$, i.e., $\boldsymbol{x}_i \in \mathcal{X}_i = \{0,1\}^{N-1}$, to request learning information denoted by $\{\mathcal{M}_j | j \in \mathcal{N}_{-i}\}$ including the observations and learning parameters and chooses the RL action $\boldsymbol{a}_i \in \mathcal{A}_i$ according to state $\boldsymbol{s}_i \in \mathcal{S}$. The goal of learning agent $i \in \mathcal{N}$ is to maximize the communication utility u_i as a weighted sum of the RL performance and the communication cost ξ_i, i.e., the number of the selected cooperative agents. Thus, the cooperative communication game model is formulated as

$$\mathcal{G} = \Big\{\mathcal{N}, \mathcal{S}, \{\mathcal{X}_i\}_{i\in\mathcal{N}}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \{u_i\}_{i\in\mathcal{N}}\Big\} \tag{12}$$

According to [33], a profile of strategies denoted by the $N \times (N-1)$ matrix $\boldsymbol{X}^* = [\boldsymbol{x}_i^*]_{1\le i\le N}$ is a Nash equilibrium of the cooperative N-player game $\mathcal{G}$, where $\forall i \in \mathcal{N}$, $\boldsymbol{x}_i^*$ is the best response to the strategies of the other $N-1$ agents denoted by $\boldsymbol{X}_{-i}^*$. That is, $\forall \boldsymbol{x}_i \in \{0,1\}^{N-1}$, we have

$$u_i\left(\boldsymbol{x}_i, \boldsymbol{X}_{-i}^*\right) \le u_i\left(\boldsymbol{x}_i^*, \boldsymbol{X}_{-i}^*\right), \forall i \in \mathcal{N} \tag{13}$$

indicating that agents have no incentive to deviate from the equilibrium $\boldsymbol{X}^*$.

B. Performance Bound

The performance bound of the MARL communication scheme is derived based on the Nash equilibrium of the cooperative communication game in (12). Without loss of generality, the communication among N learning agents is assumed to follow the informational model as proposed in [7], which analyzes the effect of the learning experience exchange on the RL performance. Specifically, the RL performance of learning agent i at each time slot increases with the total information gain obtained from the cooperative communication process denoted by $\mathcal{I}_i$, which consists of the local information gain denoted by $\mathcal{H}_{i,i}$ in local state $\boldsymbol{s}_i$ and the shared information gain from the neighboring agent $j \in \mathcal{N}_{-i}$ denoted by $\mathcal{H}_{i,j}$, and the information loss denoted by $\mathcal{L}_{i,j}$ in the experience sharing caused by the changing policies of neighboring agents. For the sake of convenience, the communication policy $\boldsymbol{x}_i$ is transformed to N-dimension vector that includes $x_{i,i} = 1$ in the i-th variable, which has no influence on the performance analysis. Thus, the total information gain $\mathcal{I}_i$ of learning agent i is given by

$$\mathcal{I}_i = \sum_{j=1}^{N} x_{i,j}\mathcal{H}_{i,j} + \big(x_{i,j} - 1\big)\mathcal{L}_{i,j} \tag{14}$$

For the learning agent i, the information gain $\mathcal{H}_{i,j}$ from the learning message $\mathcal{M}_j$ shared by neighboring agent j depends on the previous information gain from agent j denoted by $\hat{\mathcal{H}}_{i,j}$, current information fraction of agent j in the total environment information of the learning agent denoted by $C_{i,j}$, the learning function $\Lambda(\cdot)$ on the learning factor $\omega_{i,j}$ and the state $\boldsymbol{s}_j$, and the information transfer function $\Phi(\cdot)$ with property $\Phi(y) \le y$ due to the information loss. Thus, the information gain from agent j is given by

$$\mathcal{H}_{i,j} = \Lambda\left(\omega_{i,j}, \boldsymbol{s}_j\right)\Phi\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right) \tag{15}$$

Similarly, the information gain from the local state $\boldsymbol{s}_i$ is $\mathcal{H}_{i,i} = \Lambda\left(\omega_{i,i}, \boldsymbol{s}_i\right)\Phi\left(C_{i,i} - \hat{\mathcal{H}}_{i,i}\right)$.

The RL performance of learning agent i also suffers from an information loss at the agents that are chosen not to communicate with in current time slot, which results in outdated knowledge from previous shared learning messages. Specifically, the information loss $\mathcal{L}_{i,j}$ of the learning agent i at neighboring agent j with $x_{i,j} = 0$ depends on the information gain of agent j from the environment at current time slot denoted by $\mathcal{T}_j$, previous total information gain at agent j denoted by $\hat{\mathcal{I}}_j$, and previous shared information gain from agent j denoted by $\hat{\mathcal{H}}_{i,j}$, i.e.,

$$\mathcal{L}_{i,j} = \frac{\hat{\mathcal{H}}_{i,j}\mathcal{T}_j}{\hat{\mathcal{I}}_j + \mathcal{T}_j} \tag{16}$$

By (14), (15) and (16), the communication utility u_i that increases with the total information gain $\mathcal{I}_i$ and decreases with

the communication cost ξ_i is given by

$$u_i = \sum_{j\neq i}^{N} x_{i,j}\Lambda\left(\omega_{i,j}, \boldsymbol{s}_j\right)\Phi\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right) - \frac{\left(1-x_{i,j}\right)\hat{\mathcal{H}}_{i,j}\mathcal{T}_j}{\hat{\mathcal{I}}_j + \mathcal{T}_j} - \varpi^{\mathrm{C}} x_{i,j} + \Lambda\left(\omega_{i,i}, \boldsymbol{s}_i\right)\Phi\left(C_{i,i} - \hat{\mathcal{H}}_{i,i}\right) \tag{17}$$

For simplicity, the learning agent is assumed to have no information loss in the learning process, i.e., $\Phi(C_{i,j} - \hat{\mathcal{H}}_{i,j}) = C_{i,j} - \hat{\mathcal{H}}_{i,j}, \forall j \in \mathcal{N}$. The shared information from neighboring agents is assumed to have positive effect on the learning performance of learning agent i, which increases with the task similarity $\eta_{i,j}$, i.e., $\Lambda\left(\omega_{i,j}, \boldsymbol{s}_j\right) = \omega_{i,j}\eta_{i,j}$, $\forall j \in \mathcal{N}_{-i}$. Thus, by (17), the communication utility u_i is given by

$$u_i = \sum_{j\neq i}^{N} x_{i,j}\omega_{i,j}\eta_{i,j}\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right) - \frac{\left(1-x_{i,j}\right)\hat{\mathcal{H}}_{i,j}\mathcal{T}_j}{\hat{\mathcal{I}}_j + \mathcal{T}_j} - \varpi^{\mathrm{C}} x_{i,j} + \omega_{i,i}\left(C_{i,i} - \hat{\mathcal{H}}_{i,i}\right) \tag{18}$$

Theorem 1: The performance bound of MARL communication schemes in terms of the information gain $\mathcal{I}_i$, communication cost ξ_i and utility u_i, $\forall i \in \mathcal{N}$, is given by

$$\bar{\mathcal{I}}_i = \bar{\omega}\bar{\eta}\sum_{j=1,j\neq i}^{M_i}\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right) - \sum_{j=M_i+1,j\neq i}^{N}\frac{\hat{\mathcal{H}}_{i,j}\mathcal{T}_j}{\hat{\mathcal{I}}_{i,j} + \mathcal{T}_j} + \bar{\omega}\left(C_{i,i} - \hat{\mathcal{H}}_{i,i}\right) \tag{19a}$$

$$\bar{\xi}_i = M_i \tag{19b}$$

$$\bar{u}_i = \bar{\omega}\bar{\eta}\sum_{j=1,j\neq i}^{M_i}\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right) - \sum_{j=M_i+1,j\neq i}^{N}\frac{\hat{\mathcal{H}}_{i,j}\mathcal{T}_j}{\hat{\mathcal{I}}_{i,j} + \mathcal{T}_j} + \bar{\omega}\left(C_{i,i} - \hat{\mathcal{H}}_{i,i}\right) - \varpi^{\mathrm{C}} M_i, \tag{19c}$$

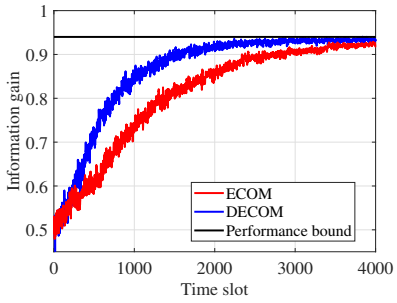
if the $N \geq M_i$ learning agents have identical learning factor $\bar{\omega}$ and task similarity $\bar{\eta}$, and $\forall j \in \mathcal{N}_{-i}$, have

$$\bar{\eta} > \frac{\varpi^{\mathrm{C}} - \hat{\mathcal{H}}_{i,j}}{\bar{\omega}\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right)} \tag{20}$$

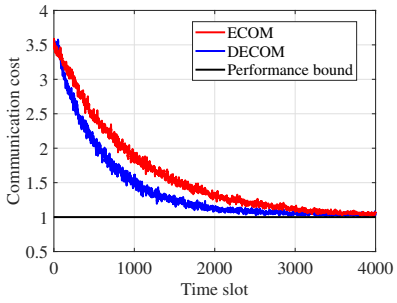
$$\frac{\max\limits_{M_i<m\leq N}\mathcal{T}_m}{\min\limits_{1\leq m\leq N}\hat{\mathcal{I}}_m} \leq \frac{\varpi^{\mathrm{C}} - \bar{\omega}\bar{\eta}\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right)}{\hat{\mathcal{H}}_{i,j} + \bar{\omega}\bar{\eta}\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right) - \varpi^{\mathrm{C}}} \leq \frac{\min\limits_{1\leq m\leq M_i}\mathcal{T}_m}{\max\limits_{1\leq m\leq N}\hat{\mathcal{I}}_m} \tag{21}$$

Proof. See Appendix A. □

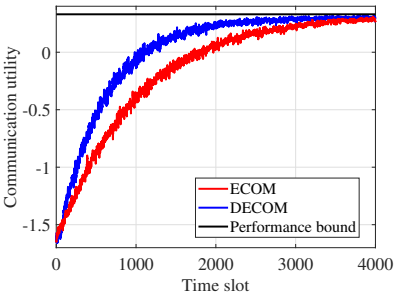
Remark 1: If the task similarity among learning agents is identical and higher than a bound that decreases with the learning factor and information fraction given by (20), learning agent i chooses to communicate with M_i neighboring agents that have more newly learned information from the environment at current time slot as given by (21). In this case, the bound of the information gain obtained from the environment and experience sharing with the neighboring agents given by (19) increases with the task similarity and the information fraction of the selected cooperative agents and decreases with the newly learnt information of the unchosen agents $\mathcal{T}_i$, $M_i + 1 \leq i \leq N$, as given by (21). Our proposed ECOM and DECOM converges to the performance bound provided in (19) after 3000 and 4000 time slots, respectively as shown in Fig. 3. For example, DECOM increases 84.0% information gain after convergence, which is about 1.1% smaller than the bound given by (19a).



(a) Information gain



(b) Communication cost



(c) Communication utility

Fig. 3. Performance of the MARL communication scheme with 8 learning agents, in which each agent chooses whether to communicate with neighboring agents for learning information sharing.

The communication energy consumption increases with the number of the selected cooperative agents and their transmit power to share the learning information. The communication cost of our proposed scheme decreases over time due to the optimization of the cooperative agents, as shown in Fig. 3. In addition, with the information gain obtained from the learning information exchange increasing, the communication cost will decrease to make a trade-off between learning and communication performance. Therefore, the total communication cost in the learning process increases with the network size N and the learning time K before convergence.

VII. Case Study: MARL-based UAV Swarm Video Transmission Against Jamming

As a case study, the proposed communication scheme can be implemented in the MARL-based anti-jamming UAV swarm transmission to improve the task performance such as PLR and energy efficiency with QoS guarantee. This work is also applicable to other MARL-based mobile applications. More specifically, at time slot k, each UAV in the swarm as the learning agent is assumed to have N neighboring UAVs and chooses the transmit channel $f^{(k)} \in \{1, 2, \ldots, F\}$ and transmit power $p^{(k)} \in [P_{\min}, P_{\max}]$ to send $b^{(k)}$-byte images or videos to the control center and support mobile applications such as traffic surveillance or target tracking. A reactive jammer applies the signal detection techniques to sense the ongoing swarm transmission and chooses the jamming channel and power to degrade the swarm transmission performance.

As an example, the RL algorithms including Q-learning and DQN are applied to choose the task action, i.e., anti-jamming transmission policy $\boldsymbol{a}^{(k)} = [p^{(k)}, f^{(k)}]$ from action set $\mathcal{A} = \{1, 2, \ldots, F\} \times \{jP_{\max}/J | 1 \le j \le J\}$ to improve the video transmission quality with less energy consumption. The proposed MARL communication scheme enables the UAV to choose the cooperative UAVs $\boldsymbol{x}^{(k)}$ for cooperative information sharing of policy learning and formulate the learning state $\boldsymbol{s}^{(k)}$ of anti-jamming policy selection to improve the transmission performance under partial observation. The shared learning parameters of cooperative UAVs are also exploited to update Q-values and neural network weights to further improve the policy optimization efficiency. This work is also applicable to other RL algorithms.

A. ECOM-enhanced UAV Swarm Transmission

At time slot k, UAV as the learning agent evaluates the message size and obtains the signal SINR received at control center ρ_1, the previous transmission delay ρ_2, the packet PLR ρ_3 and the received jamming power ς from the feedback message sent from the control center. The local observation $\boldsymbol{o}^{(k)}$ consists of the size of transmitted message b, the signal SINR received at control center ρ_1, the previous transmission delay ρ_2, the PLR ρ_3 and the received jamming power at control center ς, i.e.,

$$\boldsymbol{o}^{(k)} = \left[b, \varsigma, \boldsymbol{\rho}\right] \tag{22}$$

Based on the local observation $\boldsymbol{o}^{(k)}$, the channel state $\boldsymbol{h}$ and task similarity $\boldsymbol{\eta}$ with neighboring agents, previous W observations of neighboring UAVs $\bar{\boldsymbol{o}}$ and the number of selected agents in the last time slot, the communication learning state $\tilde{\boldsymbol{s}}^{(k)}$ is formulated as

$$\tilde{\boldsymbol{s}}^{(k)} = \left[b, \varsigma, \boldsymbol{\rho}, \boldsymbol{h}, \boldsymbol{\eta}, \bar{\boldsymbol{o}}\right], \tag{23}$$

where each element $\tilde{s}_i$ is quantized into $\tilde{q}_i$ levels, $1 \le i \le 5 + 5W + 2N$. With the quantized $\tilde{\boldsymbol{s}}^{(k)}$ as the input, the M Q-tables in the sequential architecture choose cooperative agents $\boldsymbol{x}^{(k)}$ based on $Q^{\mathrm{C}}_{1 \le j \le M}\left(\tilde{\boldsymbol{s}}^{(k)}, \cdot\right)$ via (2) to request the learning information including the observation $\hat{\boldsymbol{o}}_i$ and learning parameters $\hat{\boldsymbol{Q}}^{\mathrm{R}}_i$ of each cooperative agent $i \in \boldsymbol{x}^{(k)}$.

The state $\boldsymbol{s}^{(k)}$ of task learning, i.e., the anti-jamming transmission policy selection, consists of the local observation $\boldsymbol{o}^{(k)}$ in (22) and the encoded observations $\tilde{\boldsymbol{o}}_i$ of each cooperative agent $i \in \boldsymbol{x}^{(k)}$ based on the observation encoder, i.e.,

$$\boldsymbol{s}^{(k)} = \left[b, \varsigma, [\rho_{1 \le i \le 3}], \left[\tilde{\boldsymbol{o}}_{i \in \boldsymbol{x}^{(k)}}\right]\right] \tag{24}$$

Each element s_i in the anti-jamming policy learning state $\boldsymbol{s}^{(k)}$ is quantized into q_i levels, $1 \le i \le 5 + M$. The anti-jamming transmission policy $\boldsymbol{a}^{(k)}$ consists of the transmit channel $f^{(k)} \in \{1, 2, \ldots, F\}$ and transmit power $p^{(k)} \in \{jP_{\max}/J | 1 \le j \le J\}$ and is chosen based on the task Q-values $Q^{\mathrm{R}}(\boldsymbol{s}^{(k)}, \cdot)$ according to the ϵ-greedy method.

Upon receiving the feedback from the control center, the task reward $r^{(k)}$ is evaluated based on signal SINR ρ_1, transmission delay ρ_2, packet PLR ρ_3, UAV transmit power $p^{(k)}$ and the frequency hopping cost ϱ, given by

$$\begin{aligned} r^{(k)} =& \rho_1 - \varpi_1^{\mathrm{T}} \rho_2 - \varpi_2^{\mathrm{T}} \rho_3 - \varpi_3^{\mathrm{T}} p^{(k)} \\ & - \varrho \mathbb{I}\left(f^{(k)} \ne f^{(k-1)}\right) \end{aligned} \tag{25}$$

Based on the task reward $r^{(k)}$ and the number of selected cooperative agents ξ, the communication reward $\tilde{r}^{(k)}$ is evaluated as (7). The task Q-values are updated with $r^{(k)}$ and shared Q-values via (5) and (6). The communication Q-values are also updated with reward $\tilde{r}^{(k)}$ similar to (5).

B. DECOM-enhanced UAV Swarm Transmission

The proposed communication scheme for deep MARL is implemented in the DQN-based anti-jamming UAV swarm transmission to further enhance the performance for the high-dimension anti-jamming state space and improve the policy exploration and learning speed with shared learning observations and neural network weights from the cooperative agents. With the communication state $\tilde{\boldsymbol{s}}^{(k)}$ in (23) as the input, each level j with same network architecture, i.e., four fully-connected layers including a $(5+5W+2N)$-node input layer, two hidden layers with g_1 and g_2 nodes and an $(N+1)$-node output layer, estimates the communication Q-values $Q^{\mathrm{C}}_j\left(\tilde{\boldsymbol{s}}^{(k)}, \cdot; \boldsymbol{\alpha}_j\right)$ to choose the cooperative UAV x_j, $j \in \{1, 2, \cdots, M\}$ in the sequential architecture.

After receiving the learning response including the observations $\{\hat{\boldsymbol{o}}_i | i \in \boldsymbol{x}^{(k)}\}$ and neural network weights of DQN $\{\hat{\boldsymbol{\theta}}_i | i \in \boldsymbol{x}^{(k)}\}$ from cooperative agents $\boldsymbol{x}^{(k)}$, the anti-jamming policy transmission selection state $\boldsymbol{s}^{(k)}$ is formulated as (24) and input to the Q-network with four fully-connected layers, each having $5 + M$, ν_1, ν_2 and FJ nodes, respectively, to obtain the Q-values $Q^{\mathrm{R}}\left(\boldsymbol{s}^{(k)}, \cdot; \boldsymbol{\theta}\right)$ and choose the transmit channel and power $\boldsymbol{a}^{(k)} = [p^{(k)}, f^{(k)}]$.

Upon receiving the feedback from the control center, the anti-jamming transmission reward $r^{(k)}$ and communication reward $\tilde{r}^{(k)}$ is evaluated via (25) and (7), respectively. The anti-jamming transmission experience $\boldsymbol{z}^{\mathrm{R}}$ and communication experience $\boldsymbol{z}^{\mathrm{C}}$ are stored in the memory pool $\mathcal{D} \leftarrow \mathcal{D} \cup \{\boldsymbol{z}^{\mathrm{R}}, \boldsymbol{z}^{\mathrm{C}}\}$. The experience replay technique is used to update network weights with the stochastic gradient descent algorithm by sampling G experiences $\tilde{\boldsymbol{z}} = \left\{\boldsymbol{z}^{\mathrm{R}}, \boldsymbol{z}^{\mathrm{C}}\right\}$ to formulate the minibatch $\mathcal{B}$. The network weights are updated by minimizing

the loss function in (9) and also updated based on the shared network weights from the cooperative agents via (10).

C. Computational Complexity

The computational complexities of the proposed ECOM and DECOM increase with the number of multiplications in the cooperative agent and task policy selection and the update of the reinforcement learning layer and the communication learning layer. According to [34], the computational complexity of ECOM in the cooperative agent selection and task policy selection is $O(MN)$ and $O(FJ)$, respectively. The proposed ECOM updates the communication Q-values with computational complexity $O(MN)$ and updates task Q-values based on task reward and shared Q-values with computational complexity $O\left(FJ\prod_{i=1}^{5+M} q_i\right)$, where q_i is the quantized level of i-th element in task state. Therefore, the computational complexity of ECOM with a large anti-jamming policy space is $O\left(FJ\prod_{i=1}^{5+M} q_i\right)$.

In DECOM, M C-Networks in the sequential architecture of the communication learning layer and the task network in the reinforcement learning layer perform forward propagation and back propagation to choose the cooperative agents and anti-jamming policy and update network weights $\boldsymbol{\alpha}_{1\leq j\leq M}$ and $\boldsymbol{\theta}$. The M C-Networks in the communication learning layer perform $\phi_{\text{C,F}}$ multiplications in the forward propagation of the cooperative agent selection that linearly increases with the number of network weights in four fully-connected layers, i.e.,

$$\begin{aligned}\phi_{\text{C,F}} =&M(5+5W+2N)g_1+M(g_1+1)g_2\\&+M(g_2+1)(N+1)\end{aligned}\tag{26}$$

The observation encoder performs $\phi_{\text{E,F}}=11V$ multiplications in the forward propagation to extract correlation between local and shared observations. The task network performs forward propagation to choose the anti-jamming policy and has $\phi_{\text{T,F}}$ multiplications given by

$$\phi_{\text{T,F}}=(M+5)(\nu_1+1)+\nu_1(\nu_2+1)+(\nu_2+1)FJ\tag{27}$$

The C-Networks and task network perform forward propagation to estimate the target value in the loss function to update the network weights $\boldsymbol{\alpha}_{1\leq j\leq M}$ and $\boldsymbol{\theta}$ with $2\phi_{\text{C,F}}$ and $2\phi_{\text{T,F}}$ multiplications, respectively. To calculate the gradient the loss function and update neural network weights, M C-Networks perform $\phi_{\text{C,B}}$ multiplications in the back propagation given by

$$\begin{aligned}\phi_{\text{C,B}} =&2M(5+5W+2N)g_1+2M(g_1+1)g_2\\&+3M(g_2+1)(N+1)\end{aligned}\tag{28}$$

Similarly, the number of the multiplications in the back propagation of task network is given by

$$\begin{aligned}\phi_{\text{T,B}} =&2(M+5)(\nu_1+1)+2\nu_1(\nu_2+1)\\&+3(\nu_2+1)FJ\end{aligned}\tag{29}$$

In addition, the task network also takes $M\phi_{\text{T,F}}$ multiplications to update local neural network weights based on the shared information. Thus, the total number of multiplications of DECOM with G-experience minibatch per time slot is

$$\begin{aligned}\psi =&(2G+1)\phi_{\text{C,F}}+(2G+M+1)\phi_{\text{T,F}}+\phi_{\text{E,F}}\\&+G(\phi_{\text{C,B}}+\phi_{\text{T,B}})\end{aligned}\tag{30}$$

Theorem 2: The computational complexity of the ECOM and DECOM-based anti-jamming UAV swarm transmission scheme is given by $O\left(FJ\prod_{i=1}^{5+M} q_i\right)$ and $\mathcal{O}(GKFJ)$, respectively.

Proof. See Appendix B. □

Remark 2: The computational complexity of ECOM increases with quantized level of observation $q_{1\leq i\leq M+5}$ and transmit power level J as well as the number of selected cooperative agents M that determines the task state size. The computational complexity of DECOM depends on the number of transmit channels F and the number of quantized power level J that determine the task action space size as well as the batch size G and the number of learning samples K.

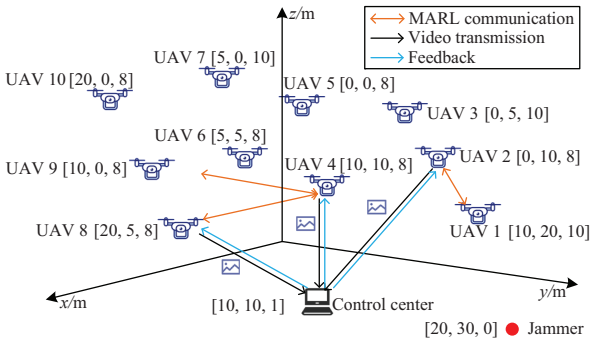
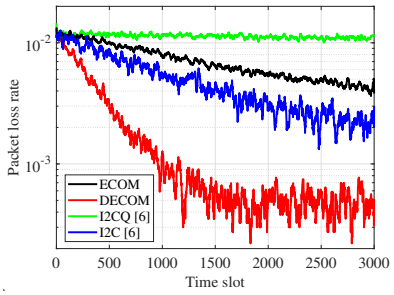


Fig. 4. Simulation setting of the swarm anti-jamming video transmission, in which each UAV sends 2-Mb videos or images to the control center with up to 100 mW transmit power and chooses the cooperative agents from the neighboring UAVs for observation and learning information sharing to accelerate anti-jamming transmission policy learning.

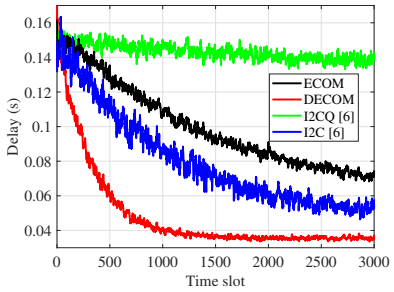
VIII. Simulation Results

The performance is evaluated in the DQN-based UAV swarm transmission against jamming attacks as shown in Fig. 4 compared with benchmark I2C in [6]. Each UAV sends 2-Mb video data to the control center modulated with quadrature phase-shift keying and chooses the transmit power with up to 100 mW on 13 channels at 2.4 GHz each with the bandwidth of 10 MHz according to 802.11n [35], [36]. The jammer sends Gaussian signals with jamming power chosen from $\{20j|0\leq j\leq 5\}$ mW at 2.4 GHz with jamming bandwidth of 30 MHz to block 3 UAV channels of swarm transmission according to the jamming model in [37].

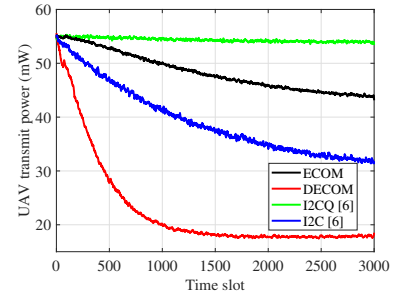
The C-Network in the communication learning layer and the task network in the task learning layer are instantiated as the four-layer neural networks including the input and output layer and two hidden layers each with 128 neural nodes. The UAV transmit power level $J=10$, the learning rate of the Q-value update $\lambda=0.4$, the discount factor of reward $\gamma=0.3$, and the greedy parameter ϵ decays from 1 to 0.01 after 1000 time slots



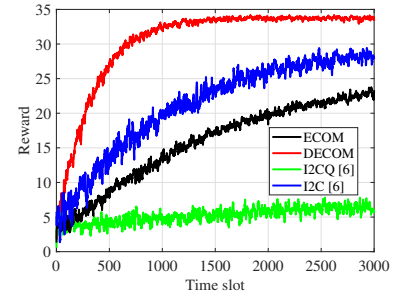
(a) Packet loss rate



(b) Delay

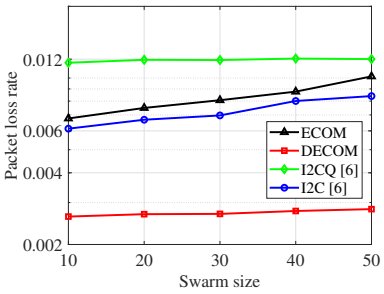


(c) UAV transmit power

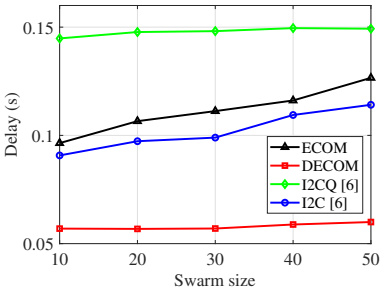


(d) Reward

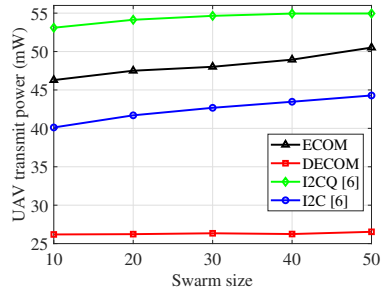
Fig. 5. Performance of the 10-UAV swarm video transmission against a jammer as shown in Fig. 4.



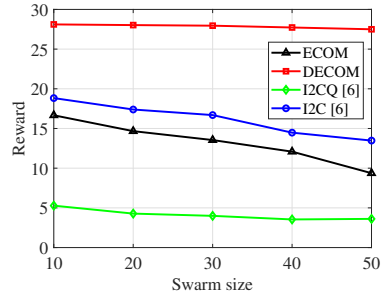
(a) Packet loss rate



(b) Delay



(c) UAV transmit power

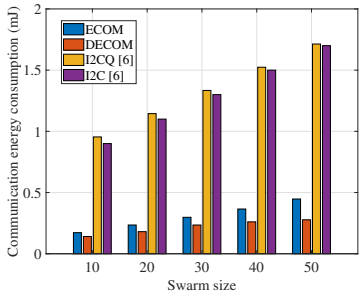


(d) Reward

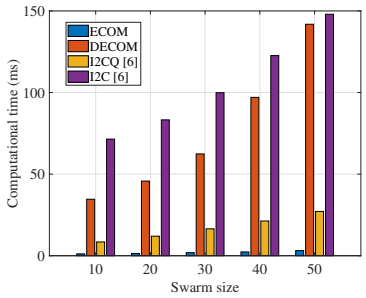
Fig. 6. Average performance of the swarm video transmission against jamming attack with different swarm sizes.

to make a trade-off between exploration and exploitation. In addition, the learning rate of Adma optimizer for the update of neural network weights $\beta = 0.01$ and the number of sampled experiences in the minibatch $G = 64$.

Compared with the benchmark MARL communication scheme I2C in [6], our proposed communication schemes accelerate the learning speed and improve the RL-based anti-jamming transmission performance as shown in Fig. 5. For example, our proposed ECOM reduces 62.5% PLR to 4.2×10^{-3} with 18.9% less transmit power compared with I2C-based Q-learning algorithm named I2CQ after 3000 time slots. The reason is that learning based communication mechanism enables learning agent to choose the most relevant cooperative agent for observation sharing and thus reduces the information redundancy in the task policy selection. Our proposed communication scheme for deep MARL further improves the performance with 80.1% reduced PLR to 4.3×10^{-4} and 43.8% less UAV transmit power after 2000 time slots and



(a) Communication energy consumption



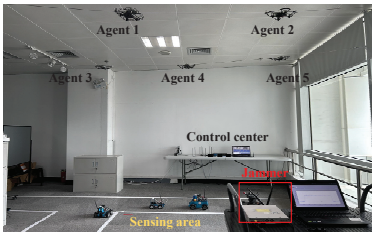
(b) Computational time

Fig. 7. MARL communication energy consumption and computational time for the UAV swarm ranging from 10 to 50 averaged over 2000 time slots.

saves 44.0% learning time compared with I2C-based DQN algorithm, because that the neural network in the communication layer accelerates the cooperative UAV selection and the shared network weights further improves the anti-jamming policy optimization.

The average performance as shown in Fig. 6 verifies the performance gain and the robustness of our proposed schemes for the swarm ranging between 10 and 50 UAVs. For example, compared with benchmark I2CQ, our proposed ECOM decreases 32.2% PLR and 25.0% transmission delay for the 30-UAV swarm, due to the adaptive cooperative agent selection that results in more efficient anti-jamming policy learning. DECOM further reduces 61.4% PLR, 42.4% transmission delay and saves 38.4% UAV transmit power and the performance gain increases with the swarm size to 66.6%, 47.4% and 40.2%, respectively, because the neural network based sequential architecture accelerates the cooperative agent selection under the large swarm size.

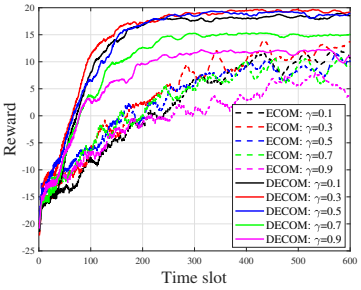
As shown in Fig. 7, our proposed schemes also improve the multi-agent communication efficiency and decrease the computational time of the communication and task learning under different swarm size compared with benchmark I2C. For example, our proposed ECOM reduces larger than 70.4% MARL communication energy consumption, which is less than 0.5 mJ for the UAV swarms ranging from 10 to 50, and 86.6% computational time due to the adaptive cooperative agent selection that reduces the information redundancy in the RL state formulation. In addition, our proposed DECOM reduces larger that 83.7% communication energy consumption compared with I2C due to the neural networks that accelerate the cooperative agent learning.



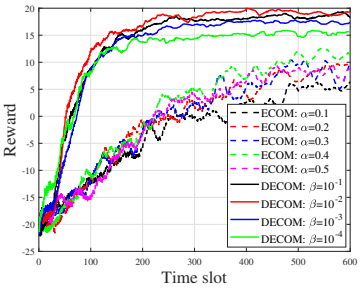
(a) Experiment setting

(b) Graphical user interface

Fig. 8. Experimental setting and graphical user interface of the UAV swarm transmission against jamming, in which each of 5 UAVs as the learning agent sends captured videos of the sensing area to the control center with transmit power up to 100 mW.



(a) Reward vs. discount factor

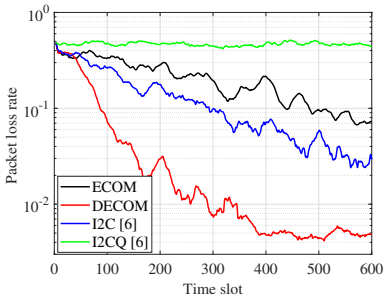


(b) Reward vs. learning rate

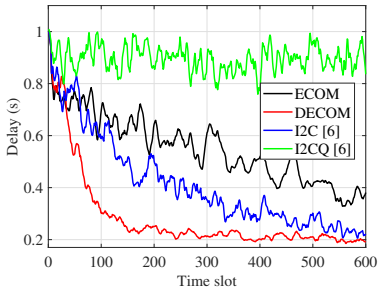
Fig. 9. Convergence performance of the anti-jamming video transmission of the proposed schemes for 5 UAVs as shown in Fig. 8 versus discount factor and learning rate averaged over 100 runs.

IX. Experimental Results

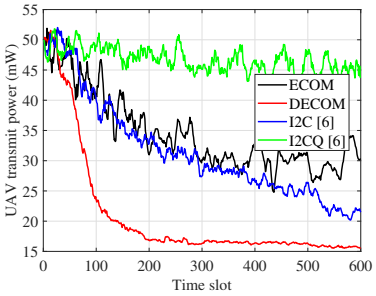
Experiments were performed to evaluate the performance gain of the proposed communication scheme in the 5-UAV swarm anti-jamming video transmission with topology as shown in Fig. 8. Each UAV equipped with Raspberry Pi sends 1-Mb captured videos of the sensing area to the control center



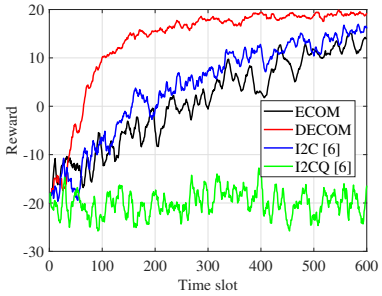
(a) Packet loss rate



(b) Delay



(c) UAV transmit power



(d) Reward

Fig. 10. Performance of the swarm video transmission to the control center against jamming attack with topology as shown in Fig. 8, in which each UAV chooses a cooperative agent from the other 4 UAVs to share observations for faster learning.

per second with up to 100 mW transmit power from one of the 6 channels at 2.4 GHz each with bandwidth 20 MHz according to IEEE 802.11n [35]. A laptop with an Intel i7-11800H 8 cores at 2.3 GHz CPU as the control center can switch the MARL-based anti-jamming transmission policy learning algorithms and display the real-time transmission performance in the graphical user interface. The jammer, i.e., a USRP N210

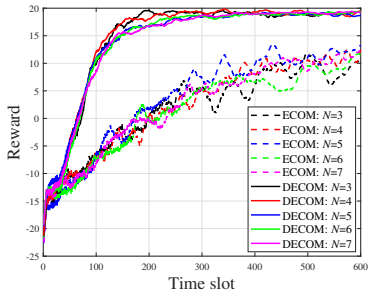
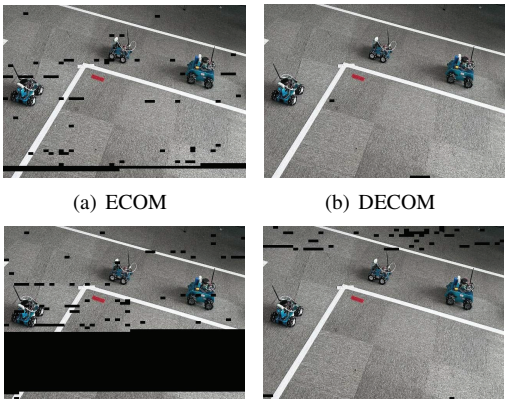


Fig. 11. Convergence performance of the anti-jamming video transmission for the swarm with $N \in [3, 7]$ UAVs averaged over 100 runs in the experimental setting as shown in Fig. 8.



(a) ECOM (b) DECOM

(c) I2CQ [6] (d) I2C [6]

Fig. 12. Anti-jamming transmission performance of UAV 5 to the control center after 500 time slots as shown in Fig. 8.

controlled by a laptop, sends Gaussian signals with up to 100 mW jamming power at center frequency 2.412 GHz with bandwidth 20 MHz.

As shown in Fig. 9, experimental results of the UAV swarm anti-jamming video transmission with topology in Fig. 8 are provided to show the effect of the discount factor and learning rate on the convergence of our proposed ECOM and DECOM. The convergence performance averaged over 100 runs under different discount factor $0.1 \leq \gamma \leq 0.9$ shows that the discount factor $\gamma = 0.3$ makes a tradeoff between the exploration and exploitation in the learning process. In addition, the learning rate $\lambda = 0.4$ of the Q-value update and the learning rate $\beta = 0.01$ in the Adam optimizer of the neural network weight update have been shown to improve the anti-jamming video transmission performance.

The performance gain over the benchmark scheme I2C in [6] is also verified via experimental results as shown in Fig. 10. For example, our proposed ECOM decreases 84.5% PLR, 55.4% transmission delay and 33.5% UAV transmit power after 500 time slots compared with I2CQ. The reason is that the adaptive cooperative agent selection chooses the agent with similar learning task in the dynamic swarm compared with the pre-trained prior network. In addition, the neural network based sequential architecture in DECOM further accelerates the communication policy learning and improves the video

transmission performance with 81.4% PLR, 18.7% transmission delay and 26.8% transmit power reduced compared with the benchmark I2C. Experimental results as shown in Fig. 11 also show that our proposed schemes are robust against the swarm size ranging from 3 to 7. In addition, both our proposed ECOM and DECOM improve the anti-jamming transmission quality with lower packet PLR compared with I2C as shown in Fig. 12 due to the MARL with enhanced communication scheme that accelerates the UAV anti-jamming transmission policy learning.

X. Conclusion

In this paper, we have proposed an efficient communication scheme to choose the cooperative agents to share the environment observations and learning experiences for MARL-based mobile applications. The correlation with the observations of the neighboring learning agents extracted based on the attention mechanism is exploited in the task state formulation to reduce the redundancy of the shared observations and enhance the RL policy exploration speed. Both the shared Q-values of state-action pairs and neural network weights are used in the policy distribution and task network update to accelerate RL task policy learning. The performance bound regarding the utility of the learning agent increases with the task similarity and the shared information of the selected agents. The communication enhanced multi-agent RL has been implemented in the UAV swarm video transmission to choose the transmit channel and power against jamming. Both the simulation and experimental results show the performance gain over the benchmark I2C, e.g., with PLR, transmission delay, transmit power and learning time reduced by 81.4%, 18.7%, 26.8% and 50.0%, respectively, in the experimental results of the 5-UAV swarm.

Appendix A
Proof of Theorem 1

Proof: By (18), $\forall x_{i,j} \in \{0,1\}$ and $j \in \{1,2,\cdots,M_i\}$, we have

$$\begin{aligned} & u_i\Big(\left[x_{i,j}, \boldsymbol{x}^*_{i,-j}\right], \boldsymbol{X}^*_{-i}\Big) \\ =& y_{i,j}x_{i,j} + \omega_0\left(C_{i,i} - \hat{\mathcal{H}}_{i,i}\right) - \frac{\hat{\mathcal{H}}_{i,j}\mathcal{T}_j}{\hat{\mathcal{I}}_j + \mathcal{T}_j} - \varpi^{\mathrm{C}}(M_i - 1) \\ & + \sum_{j\neq i}^{M_i} \bar{\omega}\bar{\eta}\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right) - \sum_{j=M_i+1}^{N} \frac{\hat{\mathcal{H}}_{i,j}\mathcal{T}_j}{\hat{\mathcal{I}}_j + \mathcal{T}_j}, \end{aligned} \tag{31}$$

where

$$\boldsymbol{x}^*_i = [\underbrace{1,1,\cdots,1}_{M_i},\underbrace{0,0,\cdots,0}_{N-M_i}] \tag{32}$$

$$y_{i,j} = \bar{\omega}\bar{\eta}\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right) + \frac{\hat{\mathcal{H}}_{i,j}\mathcal{T}_j}{\hat{\mathcal{I}}_j + \mathcal{T}_j} - \varpi^{\mathrm{C}} \tag{33}$$

By (33), (20) and (21), we have

$$y_{i,j} = \frac{\left(\bar{\omega}\bar{\eta}\left(C_{i,j} - \hat{\mathcal{H}}_{i,j}\right) - \varpi^{\mathrm{C}}\right)\left(\hat{\mathcal{I}}_j + \mathcal{T}_j\right) + \hat{\mathcal{H}}_{i,j}\mathcal{T}_j}{\hat{\mathcal{I}}_j + \mathcal{T}_j} \geq 0 \tag{34}$$

Thus, by (31) and (34), for $x_{i,j} \in \{0,1\}$ and $j \in \{1,2,\cdots,M_i\}$, we have

$$u_i\Big(\left[x_{i,j}, \boldsymbol{x}^*_{i,-j}\right], \boldsymbol{X}^*_{-i}\Big) \leq u_i\Big(\boldsymbol{x}^*_i, \boldsymbol{X}^*_{-i}\Big) \tag{35}$$

Similarly, by (18) and (21), for $x_{i,j} \in \{0,1\}$ and $\forall j \in \{M_i+1,\cdots,N\}$, we have $y_{i,j} \leq 0$ and

$$u_i\Big(\left[x_{i,j}, \boldsymbol{x}^*_{i,-j}\right], \boldsymbol{X}^*_{-i}\Big) \leq u_i\Big(\boldsymbol{x}^*_i, \boldsymbol{X}^*_{-i}\Big) \tag{36}$$

By (35) and (36), $\forall \boldsymbol{x}_i \in \{0,1\}^N$, we have

$$u_i\left(\boldsymbol{x}_i, \boldsymbol{X}^*_{-i}\right) \leq u_i\left(\boldsymbol{x}^*_i, \boldsymbol{X}^*_{-i}\right), \forall i \in \mathcal{N} \tag{37}$$

Thus, by (37), $\boldsymbol{X}^* = [\boldsymbol{x}^*_i]_{1\leq i\leq N}$ with $\boldsymbol{x}^*_i$ given by (32) is a Nash equilibrium of the game $\mathcal{G}$ and the performance bound of MARL communication scheme is given by (19) according to [33].

Appendix B
Proof of Theorem 2

Proof: According to [38], the numbers of the nodes in two hidden layers depend on the number of nodes in the output layer and the number of learning samples K. Specifically, the number of nodes in the second hidden layer $g_2 = \sqrt{K(N+1)}$. The number of nodes in the first hidden layer g_1 increases with the g_2 and the number of learning samples K, i.e.,

$$g_1 = \sqrt{K(N+1)} + 2\sqrt{\frac{K}{N+1}} \tag{38}$$

Similarly, the numbers of nodes in the hidden layers of the task network are $\nu_2 = \sqrt{KFJ}$ and $\nu_1 = \nu_2 + 2\sqrt{K/(FJ)}$.

By (26)-(30) and (38), the computational complexity of DECOM is given by

$$\begin{aligned} \Upsilon =& \mathcal{O}\big((2G+1)\phi_{\mathrm{C,F}} + (2G+M+1)\phi_{\mathrm{T,F}} \\ & + G(\phi_{\mathrm{C,B}} + \phi_{\mathrm{T,B}}) + \phi_{\mathrm{E,F}}\big) && (39) \\ =& \mathcal{O}\big(G\nu_1\nu_2 + GFJ\nu_2 + Gg_1g_2 + GNg_1 + GNg_2\big) && (40) \\ =& \mathcal{O}\left(G\sqrt{KFJ}\left(\sqrt{\frac{K}{FJ}} + \sqrt{KFJ} + FJ\right) + GN\sqrt{KN}\right. \\ & \left. + \left(G\sqrt{\frac{K}{N}} + GN\right)\left(\sqrt{KN} + 2\sqrt{\frac{K}{N}}\right)\right) && (41) \\ =& \mathcal{O}\big(GKFJ\big), && (42) \end{aligned}$$

where (40) is obtained by (26)-(29), (41) is obtained by (38), and (42) is obtained as $K \gg FJ > N$.

References

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

[2] N. Zhao, Z. Ye, Y. Pei *et al.*, "Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6949–6960, Sep. 2022.

[3] J. Foerster, I. A. Assael, N. De Freitas *et al.*, "Learning to communicate with deep multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Barcelona, Spain, Dec. 2016, pp. 2145–2153.

[4] A. Das, T. Gervet, J. Romoff *et al.*, "TarMAC: Targeted multi-agent communication," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, Jun. 2019, pp. 1538–1546.

[5] N. Naderializadeh, J. J. Sydir, M. Simsek *et al.*, "Resource management in wireless networks via multi-agent deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3507–3523, Jun. 2021.

[6] Z. Ding, T. Huang, and Z. Lu, "Learning individually inferred communication for multi-agent cooperation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2020, pp. 22 069–22 079.

[7] F. Hu, Y. Deng, and A. Hamid Aghvami, "Scalable multi-agent reinforcement learning for dynamic coordinated multipoint clustering," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 101–114, Jan. 2023.

[8] Y. Yu, S. C. Liew, and T. Wang, "Multi-agent deep reinforcement learning multiple access for heterogeneous wireless networks with imperfect channels," *IEEE Trans. Mobile Comput.*, vol. 21, no. 10, pp. 3718–3730, Oct. 2022.

[9] C. Dai, K. Zhu, and E. Hossain, "Multi-agent deep reinforcement learning for joint decoupled user association and trajectory design in full-duplex multi-UAV networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 6056–6070, Oct. 2023.

[10] H. Yang, J. Zhao, K.-Y. Lam *et al.*, "Distributed deep reinforcement learning-based spectrum and power allocation for heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6935–6948, Sep. 2022.

[11] J. Hu, H. Zhang, L. Song *et al.*, "Cooperative Internet of UAVs: Distributed trajectory design by multi-agent deep reinforcement learning," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6807–6821, Nov. 2020.

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

[13] A. A. Khan and R. S. Adve, "Centralized and distributed deep reinforcement learning methods for downlink sum-rate optimization," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8410–8426, Dec. 2020.

[14] Y. Xiao, L. Xiao, K. Wan *et al.*, "Reinforcement learning based energy-efficient collaborative inference for mobile edge computing," *IEEE Trans. Commun.*, vol. 71, no. 2, pp. 864–876, Feb. 2023.

[15] X. Zhang, H. Zhao, J. Wei *et al.*, "Cooperative trajectory design of multiple UAV base stations with heterogeneous graph neural networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 3, pp. 1495–1509, Mar. 2023.

[16] R. Zhong, X. Liu, Y. Liu *et al.*, "Multi-agent reinforcement learning in NOMA-aided UAV networks for cellular offloading," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1498–1512, Mar. 2022.

[17] X. Liu, G. Chuai, X. Wang *et al.*, "QoE-driven antenna tuning in cellular networks with cooperative multi-agent reinforcement learning," *IEEE Trans. Mobile Comput.*, pp. 1–15, Dec. 2022.

[18] Z. Lv, L. Xiao, Y. Du *et al.*, "Multi-agent reinforcement learning based UAV swarm communications against jamming," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9063–9075, Apr. 2023.

[19] S. Sukhbaatar and R. Fergus, "Learning multiagent communication with backpropagation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Barcelona, Spain, Dec. 2019, pp. 2252–2260.

[20] T. Lin, J. Huh, C. Stauffer *et al.*, "Learning to ground multi-agent communication with autoencoders," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2021, pp. 15 230–15 242.

[21] W. Kim, J. Park, and Y. Sung, "Communication in multi-agent reinforcement learning: Intention sharing," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Long Beach, CA, Apr. 2020, pp. 1538–1546.

[22] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Montréal, Canada, Dec. 2018, pp. 2252–2260.

[23] S. Q. Zhang, Q. Zhang, and J. Lin, "Efficient communication in multi-agent reinforcement learning via variance based control," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, Canada, Dec. 2019, pp. 3235–3244.

[24] R. Wang, X. He, R. Yu *et al.*, "Learning efficient multi-agent communication: An information bottleneck approach," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2020, pp. 9908–9918.

[25] T.-Y. Tung, S. Kobus, J. P. Roig *et al.*, "Effective communications: A joint learning and communication framework for multi-agent reinforcement learning over noisy channels," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2590–2603, Aug. 2021.

[26] H. Mao, Z. Zhang, Z. Xiao *et al.*, "Learning agent communication under limited bandwidth by message pruning," in *Proc. the AAAI Conference on Artificial Intelligence*, New York, NY, Apr. 2020, pp. 5142–5149.

[27] Y. Wang, J. Xu, Y. Wang *et al.*, "ToM2C: Target-oriented multi-agent communication and cooperation with theory of mind," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2022.

[28] Z. Lv, Y. Du, Y. Chen *et al.*, "Efficient communications for multi-agent reinforcement learning in wireless networks," in *Proc. IEEE Global Commun. Conf. (Globecom)*, Kuala Lumpur, Malaysia, Dec. 2023, pp. 583–588.

[29] A. Narayan and T. Y. Leong, "Effects of task similarity on policy transfer with selective exploration in reinforcement learning," in *Proc. Int. Conf. Autonomous Agents and Multiagent Systems*, Montreal, Canada, May 2019, pp. 2132–2134.

[30] Q. Zhou, Y. Niu, P. Xiang *et al.*, "Intra-domain knowledge reuse assisted reinforcement learning for fast anti-jamming communication," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4707–4720, June 2023.

[31] A. Zhang, R. T. McAllister, R. Calandra *et al.*, "Learning invariant representations for reinforcement learning without reconstruction," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vienna, Austria, May. 2021, pp. 1–12.

[32] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, Dec. 2017, pp. 6000–6010.

[33] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of reinforcement learning and control*, pp. 321–384, Jun. 2021.

[34] C. Jin, Z. Allen-Zhu, S. Bubeck *et al.*, "Is Q-learning provably efficient?" in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, Dec. 2019, pp. 1–11.

[35] *LATITUDE MATRICE 200 V2 SERIES technical parameters*, Accessed: Feb. 15, 2024. [Online]. Available: https://www.dji.com/cn/matrice-200-series-v2/info\#specs

[36] X. Lu, L. Xiao, G. Niu *et al.*, "Safe exploration in wireless security: A safe reinforcement learning algorithm with hierarchical structure," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 732–743, 2022.

[37] M. K. Hanawal, M. J. Abdel-Rahman, and M. Krunz, "Joint adaptation of frequency hopping and transmission rate for anti-jamming wireless systems," *IEEE Trans. Mobile Comput.*, vol. 15, no. 9, pp. 2247–2259, Sep. 2016.

[38] G.-B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 274–281, Mar. 2003.

Zefang Lv (Student Member, IEEE) received her B.S. degree in statistics from Shandong University in 2016 and her M.S. degree in applied statistics from North China Electric Power University in 2020. She is currently pursuing a Ph.D. degree with the Department of Informatics and Communication Engineering, Xiamen University. Her research interests include network security, wireless communications and reinforcement learning.

Liang Xiao (Senior Member, IEEE) received the B.S. degree in communication engineering from the Nanjing University of Posts and Telecommunications, China, in 2000, the M.S. degree in electrical engineering from Tsinghua University, China, in 2003, and the Ph.D. degree in electrical engineering from Rutgers University, NJ, USA, in 2009. She was a Visiting Professor with Princeton University, Virginia Tech, and the University of Maryland, College Park. She is currently a Professor with the Department of Informatics and Communication Engineering, Xiamen University, Xiamen, China. She was a recipient of the Best Paper Award for 2016 INFOCOM Big Security WS and 2017 ICC. She has served as an Associate Editor for *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY* and a Guest Editor for *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING*.

Yousong Du received the B.S. degree in Electronic Information Engineering from Beijing University of Chemical Technology, China, in 2021. He is currently pursuing the M.S. degree in the Department of Informatics and Communication Engineering, Xiamen University. His research interests include network security and wireless communications.

Yunjun Zhu received the B.S. degree in Electrical Engineering and Automation from South China University of Technology, China, in 2021. He is currently pursuing a Master's degree in the Department of Informatics and Communication Engineering at Xiamen University, China. His research interests include network security and wireless communications.

Shuai Han (S'11–M'12–SM'17) is currently a full Professor at the Department of Electronics and Communication Engineering, Harbin Institute of Technology. Shuai Han's research interests include wireless communications, satellite IoT and integrated satellite-terrestrial communication networks. Over his academic career, his students and he have contributed in various fields in wireless networks.

He has authored or co-authored over 100 technical papers in major journals and conferences. As PI, he has more than twenty grants on wireless networks and positioning. He is an associate editor of *IEEE China Communications*, *IEEE ACCESS*, *Journal of Communications and Information Networks (JCIN)*, *Journal of Telemetry*, *Tracking and Command*, *Journal of Signal Processing*. And has served as guest editor for many IEEE magazines and journals. He has served as a co-chair for technical symposia of international conference, *IEEE GC 2023*, *ICC 2023*, *IEEE GC 2021*, *IEEE GC 2019*, *IEEE ICC 2018*, *IEEE VTC FALL 2016*. He has also served as the TPC Chair for some international conferences, including the *AICON2019* and *MLICOM2018*. He is a member of 2020-2021 R10 Awards & Recognition Committee. Also, he is a senior member of *IEEE Communication Society*, Chair of IEEE BTS Chapter, Vice Chair of *IEEE Harbin ComSoc Chapter*, Vice Chair of *IEEE Harbin VTS Chapter*, and Vice Chair of *IEEE IoT-AHSN TC*.

Shuai Han began his university studies in Communication Engineering in 2000 at the Harbin Institute of Technology. He received his ME and Ph.D. degrees in Information and Communication Engineering from the Harbin Institute of Technology in 2007 and 2011, respectively. And he completed his post-doctoral work in 2012 in Electrical and Computer Engineering at Memorial University of Newfoundland in Canada.

Yong-Jin Liu (Senior Member, IEEE) is a full professor with the Department of Computer Science and Technology, Tsinghua University, China. He received the BEng degree from Tianjin University, China, in 1998, and the PhD degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2004. His research interests include machine learning, cognitive computation, computer graphics and computer-aided design. For more information, visit https://cg.cs.tsinghua.edu.cn/people/ Yongjin/Yongjin.htm