Learning-Aided UAV-Cooperation Reduces the Age-of-Information in Wireless Networks

Binqiang Chen, Dong Liu, Jianglong Zhang and Lajos Hanzo

***Abstract*—Unmanned aerial vehicles (UAVs) can enhance data collection for ground sensing nodes (SNs). Given the modest battery capacity of UAVs and the limited communication range of SNs, it is crucial to conceive efficient trajectory coordination for UAVs. However, existing studies simply decouple the joint trajectory planning policy of multiple UAVs into independent local policies, preventing their cooperation and hence limits the performance. Inspired by the observation that sharing messages among agents can promote their cooperation, we investigate the communication-assisted decentralized trajectory planning policy of multi-UAV wireless networks. Our goal is to minimize the overall energy consumption of UAVs and the average age of information of all SNs. To harness the encoded messages for learning a sophisticated policy, we conceive a communication-assisted distributed training and execution framework, and propose a communication-aided decentralized trajectory control algorithm. Our simulation results show that the proposed algorithm substantially outperforms the state-of-the-art deep reinforcement learning based methods, at a modest communication overhead.**

***Index Terms*—Multi-agent reinforcement learning, UAV, trajectory planning, AoI**

I. Introduction

Data collection via unmanned aerial vehicles (UAVs) from ground sensor nodes (SNs) is an increasingly important application. Compared to traditional base stations (BSs), UAVs are able to move close to SNs and exploit the resultant line-of-sight (LoS) channel, which reduces the transmission energy dissipation of SNs and potentially improves the freshness of data by providing flexible access for the SNs.

Age of information (AoI) is a metric widely adopted for quantifying data freshness from the receiver's perspective [1], which is critical in UAV-aided wireless networks [2, 3]. In order to minimize the average peak AoI, Abd-Elmagid *et al.* [2] investigated the UAV-assisted mobile relay problem and designed an iterative algorithm for jointly optimizing both the flight trajectory and resource allocation for packet transmissions. Liao *et al.* [3] applied a successive convex approximation based algorithm for optimizing the UAV's flight trajectory, while aiming for minimizing the average AoI and the energy consumption. However, the formulation of conventional optimization problems requires accurate and tractable models of UAV-aided wireless networks, which are not known *a priori* in practice. Moreover, the computational complexity of conventional optimization methods escalates exponentially with the growth of the number of UAVs and SNs.

Given the recent advances in deep reinforcement learning (DRL), the authors of [4–8] proposed techniques based on DRL for resolving the above challenges for AoI-optimal transmission policies. Li *et al.* [4] considered discretized trajectories and harnessed deep Q-network (DQN) to design their control policies. Fan *et al.* [5] proposed a soft actor-critic algorithm for minimizing the AoI by optimizing the trajectory and scheduling policies. Ferdowsi *et al.* [6] employed traditional convex optimization methods for continuous maneuvering UAVs, and utilized DQN for their scheduling policy of ground SNs. To learn actions involving both continuous and discrete variables, Hu *et al.* [7] combines DQN and DDPG to design UAVs trajectories for AoI minimization. To tackle multi-agent trajectory planning problems, Xu *et al.* [8] utilized an independent agent based Q-Learning method for minimizing the mission completion time of multi-UAV-aided data collection. However, the solutions in [4–8] learn the policy for each agent independently while treating other simultaneously-learning agents as part of the environment, which results in the non-stationary problems. To tackle this challenge, the centralized training and decentralized execution (CTDE) framework was adopted in [9, 10]. For instance, multi-agent deterministic policy gradient (MADDPG) technique [9] trains a centralized critic having access to the observations of all agents, so that the critic sees a stationary environment and guides the learning of each actor more wisely.

Nonetheless, the aforementioned methods implicitly assume "conditional independence" among different agents, implying that each agent can make decision independently and solely relies on local observation without considering the impact from other agents [11]. However, the intentions and behavior of agents have non-negligible impacts on others in reality, and thus it is critical to establish an information sharing mechanism for learning more effective coordination.

Against the above background, we propose a communication assisted decentralized training and execution framework, where the agents can share their messages via their communication channels for promoting cooperation. In particular, we study the decentralized trajectory control problem of multi-UAV-aided wireless networks, where multiple UAVs are deployed to cooperatively collect data from ground SNs. In order to minimize the weighted-average AoI and the energy dissipation of UAVs, we formulate a multi-agent reinforcement learning (MARL) problem for optimizing the trajectory of UAVs. To relax the assumption of conditional independence among agents and promote cooperation, we propose a communication-assisted decentralized trajectory control (CADTC) algorithm, where all agent make decision in a decentralized way but with the aid of messages from agents

This work was supported in part by the National Natural Science Foundation of China under Grant 62001509 and Grant 62301015; in part by the Youth Top Talent Support Program of Beihang University under Grant YWF-22-L-1269; and in part by the CAAC Key Laboratory of General Aviation Operation under Grant CAMICKFJJ-2020-4. *(Corresponding author: Dong Liu)*

Binqiang Chen, Dong Liu and Jianglong Zhang are with Beihang University, Beijing, 100191, China. (e-mail: {chenbq, dliu, zhangjl1330}@buaa.edu.cn)

Lajos Hanzo is with the School of Electronics and Computer Science, the University of Southampton, Southampton SO17 1BJ, U.K. (e-mail:lh@ecs.soton.ac.uk)

in proximity via communication. Our simulation results show that the CADTC conceived outperforms the state-of-the-art DRL-based UAV control algorithms.

II. System Model

Consider a UAV-aided wireless sensor network that consists of a base station (BS), K SNs and N rotary-wing UAVs in a region of interest. Let $\mathcal{K} = \{k|k = 1, 2, ..., K\}$ and $\mathcal{N} \triangleq \{n|n = 1, 2, ..., N\}$ denote the set of SNs and UAVs, respectively. Each SN samples data from the outdoor environment, such as air pollution status and temperature, and stores the data into its local buffer. All SNs and UAVs are equipped with GPS module to enable localization services, and are willing to share their locations for cooperatively improving system performance as in [4–8]. To conserve the SNs' energy, the UAVs are deployed as mobile relays for collecting the sampled data from the SNs and for forwarding to the BS. We consider the "full buffer" scenario, where the SN's buffer will be immediately filled by new sampled data after successfully transmitting the previous data to the UAVs.

A. UAV-SN Association Assignment

We assume that all UAVs fly at a constant altitude H as in [3]. The coordinates projected onto the horizontal plane of the kth SN and the nth UAV at time step (TS) t are denoted as $\boldsymbol{c}_{k,t}^{\mathrm{SN}} = [x_{k,t}, y_{k,t}]$ and $\boldsymbol{c}_{n,t}^{\mathrm{UAV}} = [x_{n,t}, y_{n,t}]$, respectively. Then, the nth UAV computes the horizontal distance to SN k in TS t as $d_{k,n,t} = \|\boldsymbol{c}_{k,t}^{\mathrm{SN}} - \boldsymbol{c}_{n,t}^{\mathrm{UAV}}\|$.

The maximal horizontal communication radius of SNs is denoted as $D_{\max}$. For each SN, if there exists UAVs within the SN's communication range, the SN is associated to the nearest UAV and transmits its local data. Otherwise, the SN keeps data in its local buffer. Furthermore, the *active* SNs, having at least one UAV within its communication range, are represented as $\mathcal{K}_t = \{k'|\exists n' \in \mathcal{N}, s.t., d_{k',n',t} \leq D_{\max}\}$.

B. UAV Motion Control and Energy Cost

Let us denote the maximal UAV speed by $V_{\max}$ and the normalized velocity of the nth UAV at TS t by $\boldsymbol{v}_{n,t} = [v_{x,n,t}, v_{y,n,t}]$ with $\|\boldsymbol{v}_{n,t}\| \leq 1$. The position of the nth UAV is expressed as $\boldsymbol{c}_{n,t+1} = \boldsymbol{c}_{n,t} + (\boldsymbol{v}_{n,t} V_{\max} + \boldsymbol{w}_{n,t})\delta$, where $\boldsymbol{w}_{n,t}$ represents the random effects introduced by the environment, δ is the duration of each TS, and the effects of the UAV's deceleration and acceleration are neglected as in [3],

The propulsion energy dissipation of UAV n in TS t is $C_{n,t} = E_{n,t}\delta$, where $E_{n,t} \approx P_0(1 + 3||\boldsymbol{v}_{n,t}||^2 V_{\max}^2/U_{\mathrm{tip}}^2)$ is the propulsion-based consumption per unit time, P_0 is the blade power, and U_{tip} is the tip speed of the rotor [12]. Note that the hovering cost is neglected here for focusing on the energy consumed during the movement of UAVs as in [13]. In particular, when $||\boldsymbol{v}_{n,t}|| = 1$, $C_{n,t}$ achieves its maximal value of $C_{\max} = P_0(1 + 3V_{\max}^2/U_{\mathrm{tip}}^2)\delta$.

C. Channel Model and Data Collection Process

We apply the probabilistic path loss model for characterizing the uplink channel between a SN and a UAV. The channel's path loss from the SN to the UAV at TS t is given by $h_{k,n,t} = d_{k,n,t}^{-\alpha}[\mathbb{P}_{k,n,t}^{\mathrm{LoS}}\mu_{\mathrm{LoS}} + \mathbb{P}_{k,n,t}^{\mathrm{NLoS}}\mu_{\mathrm{NLoS}}]$, where α is the path loss exponent [14]. μ_{LoS} and μ_{NLoS} are the average additional loss for the LoS and NLoS links, respectively. $\mathbb{P}^{\mathrm{LoS}}$ and $\mathbb{P}^{\mathrm{NLoS}}$ are the corresponding probability of LOS and NLOS propagation, respectively.

We assume that all SNs share the same bandwidth W to communicate with the UAVs, and the ground-to-air links are scheduled using TDMA. Then, the data rate of the nth UAV serving the kth SN at TS t can be written as

$$R_{k,n,t} = \frac{W}{|\mathcal{K}_t|}\log_2\left(1 + \frac{P_{k,t}h_{k,n,t}}{\sigma^2}\right), \tag{1}$$

where $P_{k,t}$ is the normalized transmit power of the kth SN, and $|\mathcal{K}_t|$ represents the number of active SNs at TS t. Let us denote the size of data sampled by SN k until TS t as $S(k,t)$. The expected duration of transmitting its data is $\Delta_{k,n,t} = S(k,t)/R_{k,n,t}$. Thus, if $\Delta_{k,n,t} \leq \tau$, the sampled data can be successfully transmitted to UAV n. Otherwise, the remaining data will be wait for the next TS, and $S(k,t+1) = S(k,t) - R_{k,n,t}\tau$.

D. Performance Metric and Problem Formulation

We employ the AoI for quantifying the freshness of information, which takes both the waiting time to be scheduled and the transmission delay into account. Specifically, let $U_k(t)$ denote the time instant at which the latest data is transmitted completely by SN k. Then, the AoI of SN k at the beginning of TS t can be expressed as $\delta_{k,t} = t - U_k(t)$. At TS t, if the local data of SN k is successfully transmitted to an UAV located in the coverage of SN k, then its AoI is decreased to one; otherwise, it is increased by one. Then, the dynamics of the AoI can be formulated as

$$\delta_{k,t+1} = \begin{cases} \delta_{k,t} + 1, \text{if SN } k \text{ is waiting to be scheduled} \\ \delta_{k,t} + 1, \text{if SN } k \text{ is transmitting} \\ 1, \text{if the transmission is just finished} \end{cases} \tag{2}$$

We formulate trajectory control problem as follows

$$\mathbf{P1}: \min_{\boldsymbol{v}_{n,t}} \mathbb{E}\left[\sum_{t=1}^{T}\left(\lambda\sum_{k=1}^{K}\frac{\delta_{k,t}}{K} + (1-\lambda)\sum_{n=1}^{N}\frac{C_{n,t}}{C_{\max}}\right)\right] \quad s.t.\ ||\boldsymbol{v}_{n,t}|| \leq 1, \tag{3}$$

where λ is a hyper-parameter balancing the relative importance of terms in the optimization objective of (3). The objective of the problem is to simultaneously minimize the average AoI of SNs and the energy cost of UAVs, and the constraint reflects the limitation of the UAV speed. The expectation is taken over all random variables, including the UAV locations and fading channels at each TS.

The position of users, the channel distributions, as well as the randomness factor of UAV mobility are all unknown before deploying the UAVs. Furthermore, the objective function involving the derivation of the number of active SNs $\mathcal{K}_t$ is analytically intractable, which makes traditional optimization methods unsuitable. Therefore, we apply model-free MARL techniques for solving the problem.

III. CADTC ALGORITHM

In this section, we first reformulate problem (3) in the form of MARL. Then, we establish the communication assisted fully decentralized training and execution framework and propose our CADTC algorithm.

A. MARL Problem Formulation

Problem (3) can be modeled as a multi-agent decentralized partially observable Markov decision process (Dec-POMDP), which is defined by a state space $\mathcal{S}$ of legitimate environmental status, an action space $\mathcal{A}$, and an observation space $\mathcal{O}$ for each agent [9, 11]. At each TS t, the nth agent obtains its local observation $\boldsymbol{o}_{n,t} \in \mathcal{O}$ concerning the environmental state $\boldsymbol{s}_t$, and produces an action $\boldsymbol{a}_{n,t} = \pi_n(\boldsymbol{o}_{n,t}) \in \mathcal{A}$, according to its own policy π_n based on its local observation $\boldsymbol{o}_{n,t}$. When all agents complete their actions, agent n obtains reward $r_{n,t}$, which depends on state $\boldsymbol{s}_t$ and the actions of all agents $\{\boldsymbol{a}_{n,t}|1 \le n \le N\}$. Then, state $\boldsymbol{s}_t$ transits into a new state $\boldsymbol{s}_{t+1}$. For cooperative tasks, all agents aim for maximizing the total expected return $\mathbb{E}\left[\sum_{t=1}^{T}\sum_{n=1}^{N}\gamma^{t-1}r_{n,t}\right]$, where γ is a discount factor.

In the following, we specify the local observation as well as the action and then design the reward function for each agent at TS t as follows.

1) **Observation** $\boldsymbol{o}_{n,t}$: Since the AoI depends on the distance between the UAVs and users, the observation vector of each UAV should include the coordinates of the SNs and UAVs. Considering furthermore that each SN has limited communication range, we only include the coordinates of the nearest M_{d}^{SN} SNs and the nearest M_{d}^{UAV}. Then, the observation vector of the nth UAV can be formulated as $\boldsymbol{o}_{n,t} = [\boldsymbol{c}_{1,t}^{\mathrm{SN}}, \boldsymbol{c}_{2,t}^{\mathrm{SN}}, ..., \boldsymbol{c}_{M_d^{SN},t}^{\mathrm{SN}}, \boldsymbol{c}_{1,t}^{\mathrm{UAV}}, \boldsymbol{c}_{2,t}^{\mathrm{UAV}}, ..., \boldsymbol{c}_{M_d^{UAV},t}^{\mathrm{UAV}}]$.

2) **Action** $\boldsymbol{a}_{n,t}$: The UAV agent adjusts the velocity to change its trajectory. Then, the action of the nth UAV is formulated as $\boldsymbol{a}_{n,t} = [v_{x,n,t}, v_{y,n,t}]$

3) **Reward** $r_{n,t}$: Since we consider cooperative tasks, the team reward is shared among all agents, i.e, we have $r_{n,t} = -\left(\lambda\sum_{k=1}^{K}\frac{\delta_{k,t}}{K} + (1-\lambda)\sum_{n=1}^{N}\frac{C_{n,t}}{C_{\max}}\right) - \sum_{n=1}^{N}c_n^{\mathrm{col}}$, where c_n^{col} is a penalty term introduced for avoiding collisions between the UAVs, which is set to one, when the nth UAV collides with others and it is set to zero otherwise.

Then, the policy of each agent is optimized by cooperatively minimizing the cost function, which can be expressed as $\min_{\pi_n, 1\le n\le N} J = \mathbb{E}\left[-\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N}\gamma^{t-1}r_{n,t}\right]$.

B. Communication Assisted Decentralized Reinforcement Learning Framework

The widely adopted MARL algorithms such as MADDPG [9] and the value-decomposition networks of [10] implicitly assume "conditional independence" among different agents. Consequently, the globally optimal policies of multi-agents are simply decoupled into the combination of locally optimal policies of all agents. Thus, the resultant policy of each agent only takes local observations into account, but this policy is vulnerable, because it neglects the mutual impact of all agents' actions on each other.

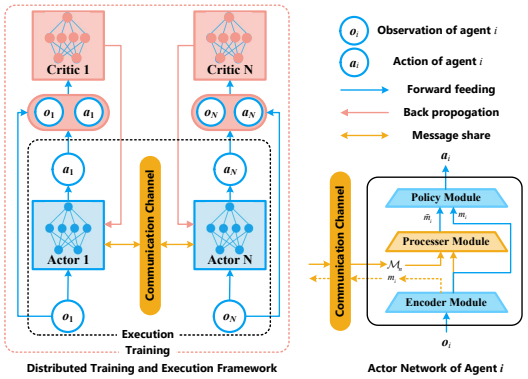

Fig. 1. Communication-assisted decentralized training and execution framework.

Inspired by recent advances in MARL, where the agents learn to share their messages via a communication channel for promoting cooperation among agents during both training and execution [15], we propose a communication-assisted actor-critic framework (CAAC) for Dec-POMDP settings, which is shown in Fig. 1(a). In order to realize decentralized training for reducing complexity, each agent has its own critic network, which takes its local observation and action as the input and outputs the estimated state-action value. In contrast to the standard MADDPG, the actor network takes the messages of others as its input in addition to its local observation. Consequently, each agent can acquire information that contains both observation and intention information of the nearby agents, which stimulates cooperation for improved decision-making.

C. Network Structure and Training Algorithm

The actor network of each agent contains three modules: the *message encoder*, the *information extraction* module, and the *policy* module, each of which is composed of artificial neural networks and the network parameters of all agents are identical. In the following, we consider agent n as an example to design the actor network in detail.

(1) The message encoder e_n takes the local observation $\boldsymbol{o}_n$ as its input, and encodes $\boldsymbol{o}_n$ into the *individual message* $\boldsymbol{m}_n$ to be shared with other agents, i.e., $\boldsymbol{m}_n = e_n(\boldsymbol{o}_n; \theta_e)$ where θ_e denotes the encoding module parameter. The message $\boldsymbol{m}_n$ contains the information about agent n's local observation. Moreover, as a hidden layer of the actor network used for generating the agent's action, $\boldsymbol{m}_n$ also contains the information concerning the action, i.e., the agent's intention. By sharing both the encoding of the local observation and action intention, the individual agents are able to build up a more global perception of the environment, infer the intentions of other agents, and cooperate on decision making.

(2) The information extraction module takes the messages of other coordinated agents as input, and outputs the *integrated message* $\widetilde{\boldsymbol{m}}_n$ that guides the agents to generate their actions. Let $\mathcal{M}_n$ denote the messages received by agent n from its M_d^{UAV} nearest agents. Then, the information extraction module can be represented as $\widetilde{\boldsymbol{m}}_n = g_n(\mathcal{M}_n, \boldsymbol{m}_n; \theta_g)$, where θ_g is the module parameter.

(3) The integrated message $\widetilde{\boldsymbol{m}}_n$ is combined with the individual message $\boldsymbol{m}_n$, and fed into the policy network, resulting in a "skip-connection" (also known as shortcut connection) of individual message as in [16]. This architecture can help the policy module to distinguish local information from the messages gleaned from other agents, and hence generate improved actions. Finally, the policy network outputs the action $\boldsymbol{a}_n = \mu_n(\boldsymbol{m}_n, \boldsymbol{m}'_n; \theta_\mu)$. Let us denote the actor network of agent n by π_n, which yields $\boldsymbol{a}_n = \pi_n(\boldsymbol{o}_n, \mathcal{M}_n; \theta_\pi) = \mu(g(e(\boldsymbol{o}_n; \theta_e), \mathcal{M}_n; \theta_g), e(\boldsymbol{o}_n); \theta_\mu)$, where we have $\theta_\pi = \{\theta_e, \theta_g, \theta_\mu\}$.

The critic of agent n is denoted as $Q_n^{\pi_n}(\boldsymbol{o}, \boldsymbol{a}; \theta_Q)$, which applies a deep neural network with parameter θ_Q to estimate the Q-value function $Q_n^{\pi_n}(\boldsymbol{o}, \boldsymbol{a}) = \mathbb{E}[\sum_{i=t}^{T} \gamma^{i-t} r_{n,t} | \boldsymbol{o}_{n,t} = \boldsymbol{o}, \boldsymbol{a}_{n,t} = \boldsymbol{a}, \pi_n]$. The critic then processes both the action as well as the observation of agent n as its input, and outputs the Q-value Q_n, as shown in Fig. 2.

The training procedure is shown in Fig. 2, which consists of the sample collection, actor update and critic update. During the sampling process, both the interactions of agent n with the environment and the messages received by agent n, are collected as a five-tuple ($\boldsymbol{o}_{n,t}$, $\mathcal{M}_{n,t}$, $\boldsymbol{a}_{n,t}$, $r_{n,t}$, $\boldsymbol{o}_{n,t+1}$) and are stored in an experience *replay buffer* $\mathcal{D}$.

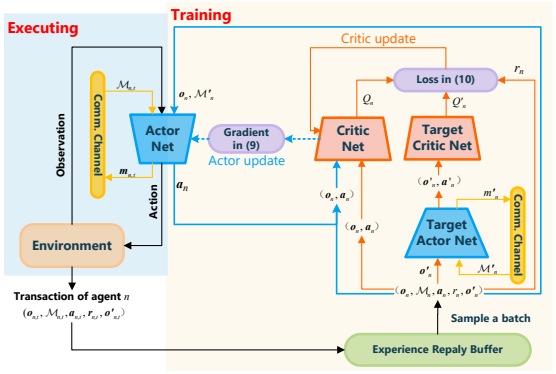

Fig. 2. The training and execution process of the nth UAV agent.

During the actor and critic update process, a set of samples $\{(\boldsymbol{o}^i, \boldsymbol{a}^i, \mathcal{M}^i, r^i, \boldsymbol{o}'^i)\}$ are sampled randomly from $\mathcal{D}$, where $1 \le i \le B$ and B is the batch size. The actor network parameter is updated for maximizing the expected return based on the policy gradient theorem. Specifically, the sampled gradient of the expected return with respect to θ_π can be expressed as

$$\nabla_{\theta_\pi} J(\theta_\pi) = \sum_i \frac{1}{B} [\nabla_{\theta_\pi} \pi_n(\boldsymbol{o}^i, \mathcal{M}^i; \theta_\pi) \nabla_{\boldsymbol{a}} Q_n^{\pi_n}(\boldsymbol{o}, \boldsymbol{a})|_{\boldsymbol{o}=o^i, \boldsymbol{a}=\pi_n(\boldsymbol{o}^i, \mathcal{M}^i; \theta_\pi)}]. \tag{4}$$

Then, the policy parameters θ_π are updated along the direction of the corresponding gradient. The critic $Q_n^{\pi_n}$ is updated as shown in Fig. 2 for minimizing the following loss function,

$$\mathcal{L}(\theta_Q) = \sum_i \frac{1}{B} [(Q_n^{\pi_n}(\boldsymbol{o}^i, \boldsymbol{a}^i; \theta_Q) - y^i)^2], \tag{5}$$

where $y^i = r^i + \gamma \widehat{Q}_n^{\widehat{\pi}_n}(\boldsymbol{o}', \boldsymbol{a}'; \widehat{\theta}_Q)|_{\boldsymbol{o}'=\boldsymbol{o}'^i, \boldsymbol{a}'=\widehat{\pi}_n(\boldsymbol{o}'^i, \mathcal{M}'^i; \widehat{\theta}_a)}$. $\widehat{\pi}_n(\boldsymbol{o}, \boldsymbol{a}; \widehat{\theta}_a)$ and $\widehat{Q}_n^{\widehat{\pi}_n}(\boldsymbol{o}, \boldsymbol{a}; \widehat{\theta}_Q)$ represent the corresponding *target networks* of the actor and critic, respectively. The detailed steps are provided in Algorithm 1.

Algorithm 1 CADTC algorithm

Randomly initialize θ_π, θ_Q, $\widehat{\theta}^\pi$ and $\widehat{\theta}^Q$.
Set $\mathcal{D} = \emptyset$.
for episode $= 1$ to max-episode-number **do**
  Reset the environment, and obtain the initial local observation $\boldsymbol{o}_n$ for $n = 1$ to N.
  for $t = 1$ to T **do**
    Get message $m_n = e(\boldsymbol{o}_n; \theta_e)$ for agent $n = 1$ to N.
    Select action $\boldsymbol{a}_n = \pi_n(\boldsymbol{o}_n, \mathcal{M}_n) +$ noise for each agent.
    Execute actions $\{\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_N\}$. Then, obtain reward r_n and next state observations $\boldsymbol{o}'_n$ for each agent n.
    Push $(\boldsymbol{o}_n, \boldsymbol{a}_n, \mathcal{M}_n, \boldsymbol{r}, \boldsymbol{o}'_n)$ into replay buffer $\mathcal{D}_n$ and set $\boldsymbol{o}_n \leftarrow \boldsymbol{o}'_n$ for $n = 1$ to N.
    if length of $\mathcal{D}_n$ larger than given length **then**
      for UAV agent $n = 1$ to N **do**
        Obtain $\{(\boldsymbol{o}^i, \boldsymbol{a}^i, \mathcal{M}^i, r^i, \boldsymbol{o}'^i)\}$, where $1 \le i \le B$.
        Update θ_π by gradient descent as in (4).
        Update θ_Q by minimizing $\mathcal{L}(\theta_Q)$ in (5).
      end for
      Update target network parameters by $\widehat{\theta} = \tau\theta + (1-\tau)\widehat{\theta}$ for $\theta \in \{\theta_\pi, \theta_Q\}$.
    end if
  end for
end for

IV. Simulation Results

In this section, we compare the performance of our DTPC algorithm to several baseline methods via simulations. To validate the effectiveness of skip-connection, we also consider the CADTC method without skip-connection in the actor network, denoted as CADTC-noSC, where the integrated message $\widetilde{\boldsymbol{m}}_n$ is directly fed into the policy network, without merging it with the individual message m_n. We also consider the following state-of-the-art DRL-based trajectory planning methods as baselines: (1) DDPG used in [7]; (2) MADDPG proposed in [9]. Moreover, we also compare our CADTC algorithm with *random*, *greedy* and *auction-based* [17] UAV trajectory control methods, where each UAV moves either randomly or consistently towards the nearest SN.

We consider a 200×200 m^2 square area containing $K = 10$ randomly distributed CNs and $N = 5$ UAVs at the altitude of $H = 100$ m. Only the SNs with a "horizontal distance" smaller than $D_{\text{th}} = 20$ m can be discovered by and connected to the UAV, and we set $M_{\text{d}}^{SN} = M_{\text{d}}^{UAV} = 2$. For the channel model, we set $\alpha = 2$, $\mu_{\text{LoS}} = -3$ dB and $\mu_{\text{NLoS}} = -23$ dB as in [14]. The noise power, maximal transmit power, and the blade power are $\sigma^2 = -100$ dBm, $P_{\max} = 500$ mW, and $P_0 = 580$ W [12], respectively. The maximal speed and rotor tip speed are $V_{\max} = 40$ m/s and $U_{\text{tip}} = 200$ m/s, respectively, as in [12]. The duration of TS $\delta = 1$ s. Detailed settings regarding the neural networks and the training process can be found at our GitHub repository: https://github.com/chenbq/CADTC.

In Fig. 3, we compare the average AoI and power consumption achieved by different methods versus the number of episodes during training. We can see that DDPG and MADDPG have similar performance, which is consistent with the conclusion in [18]. This is because both DDPG and MADPPG

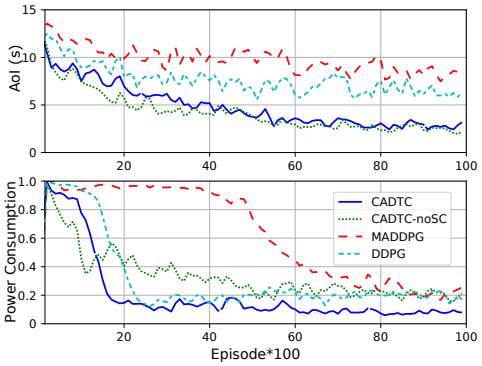

Fig. 3. Learning curves, where the AoI and power consumption are averaged over all UAV ABSs/SNs and timesteps in each episode. The unit of AoI is second. The power consumption is normalized, λ=0.5.

assume "conditional independence", and the learned policy of each agent ignores the impact of others. By contrast, CADTC-noSC and CADTC achieve lower average AoI at a lower power consumption, because the messages gleaned from communications provide extra information. This extra information introduces neglectable communication overhead, but helps agents to better understand the environment and others, thus promotes multi-agent cooperation. This justifies the necessity of communication. Furthermore, CADTC-noSC is inferior to CADTC in terms of its power consumption, which shows the benefits of employing the skip-connection architecture in the actor network.

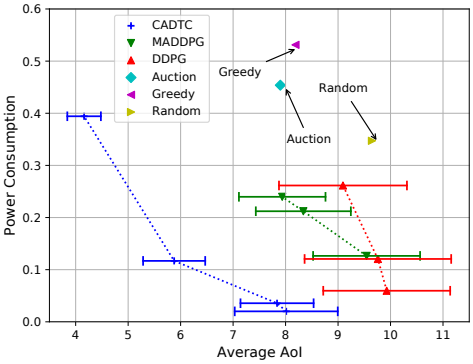

Fig. 4. Trade-off between average AoI and power consumption in the execution phase. Only points in terms of non-dominated solution are plotted for all methods. The error bar represents the 95% confidence interval.

In Fig. 4, we further compare the AoI and power consumption averaged over each episode of the different methods by varying λ in the reward function from 0 to 1. For the DRL-based methods, each point in Fig. 4 is obtained by a convergent DRL model after training with different λ. To verify the generalization capability attained, here we randomly generate the positions of SNs, which are different from those during the training phase. The curves interpolated by all the points visualize the Pareto-front of the multi-objective (average AoI and power consumption) problem in (III-A), where the Pareto-front shows that none of the objective can be improved without scarifying at least one of the other objectives. We can see that our CADTC approach dominates other baselines, both in terms of the average AoI and the power consumption. We can also see that the confidence interval for CADTC tends to be the smallest among all methods.

V. Conclusions

In this letter, we proposed a novel communication-assisted MARL based trajectory planning scheme for minimizing both the energy consumption of UAVs and the average age of information. In this scheme, agents can share messages between neighbors for enhancing cooperation in order to optimized their trajectory policies in a distributed manner. Our simulation results showed that the proposed CADTC algorithm outperforms the state-of-the-art DRL-based UAV control methods, demonstrating the power of intelligent message exchange, which allows each agent to be fully aware of the situation and of the intention of others for improved decision-making.

References

[1] S. K. Kaul, R. D. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012.

[2] M. A. Abd-Elmagid and H. S. Dhillon, "Average peak age-of-information minimization in UAV-assisted IoT networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 2003–2008, Feb. 2019.

[3] Y. Liao and V. Friderikos, "Energy and age Pareto optimal trajectories in UAV-assisted wireless data collection," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 9101–9106, Aug. 2022.

[4] S. Li, F. Wu, S. Luo, Z. Fan, J. Chen, and S. Fu, "Dynamic online trajectory planning for a UAV-enabled data collection system," *IEEE Trans. Veh. Technol.*, vol. 71, no. 12, pp. 13 332–13 343, Dec. 2022.

[5] X. Fan *et al.*, "RIS-assisted UAV for fresh data collection in 3D urban environments: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 632–647, Jan. 2023.

[6] A. Ferdowsi, M. A. Abd-Elmagid, W. Saad, and H. S. Dhillon, "Neural combinatorial deep reinforcement learning for age-optimal joint trajectory and scheduling design in UAV-assisted networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1250–1265, May 2021.

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

[8] S. Xu, X. Zhang, C. Li, D. Wang, and L. Yang, "Deep reinforcement learning approach for joint trajectory design in multi-UAV IoT networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3389–3394, Mar. 2022.

[9] R. Lowe *et al.*, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. NeurIPS*, Long Beach, CA, USA, 2017.

[10] P. Sunehag *et al.*, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proc. AAMAS*, Stockholm, Sweden, 2018.

[11] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Springer Auton. Agent Multi-Agent Syst.*, vol. 33, no. 6, pp. 750–797, Oct. 2019.

[12] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. on Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[13] Y. Sun, L. Li, Q. Cheng, D. Wang, W. Liang, X. Li, and Z. Han, "Joint trajectory and power optimization in multi-type UAVs network with mean field Q-learning," in *Proc. IEEE ICC*, Dublin, Ireland, 2020.

[14] Y. Wang *et al.*, "Three-dimensional aerial cell partitioning based on optimal transport theory," in *Proc. IEEE ICC*, Dublin, Ireland, 2020.

[15] H. Mao *et al.*, "Learning agent communication under limited bandwidth by message pruning," in *Proc. AAAI*, New York, NY, USA, 2020.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, New York, NY, USA, 2016, pp. 770–778.

[17] C. Singhal and S. De, *Resource Allocation in Next-Generation Broadband Wireless Access Networks*. PA, USA: IGI Global, 2017.

[18] X. Lyu *et al.*, "Contrasting centralized and decentralized critics in multi-agent reinforcement learning," in *Proc. AAMAS*, Virtual, 2021.