Multi-UAV Collision Avoidance using Multi-Agent Reinforcement Learning with Counterfactual Credit Assignment

多无人机避障使用带有反事实信用分配的多智能体强化学习

Shuangyao Huang, Haibo Zhang and Zhiyi Huang
汪双尧，张海滨，黄智毅

Abstract-Multi-UAV collision avoidance is a challenging task for UAV swarm applications due to the need of tight cooperation among swarm members for collision-free path planning. Centralized Training with Decentralized Execution (CTDE) in Multi-Agent Reinforcement Learning is a promising method for multi-UAV collision avoidance, in which the key challenge is to effectively learn decentralized policies that can maximize a global reward cooperatively. We propose a new multi-agent critic-actor learning scheme called MACA for UAV swarm collision avoidance. MACA uses a centralized critic to maximize the discounted global reward that considers both safety and energy efficiency, and an actor per UAV to find decentralized policies to avoid collisions. To solve the credit assignment problem in CTDE, we design a counterfactual baseline that marginalizes both an agent's state and action, enabling to evaluate the importance of an agent in the joint observation-action space. To train and evaluate MACA, we design our own simulation environment MACAEnv to closely mimic the realistic behaviors of a UAV swarm. Simulation results show that MACA achieves more than 16% higher average reward than two state-of-the-art MARL algorithms and reduces failure rate by 90% and response time by over 99% compared to a conventional UAV swarm collision avoidance algorithm in all test scenarios.

摘要-多无人机避障是无人机群应用中的一项挑战性任务，因为这需要群成员之间进行紧密合作以规划无碰撞路径。在多智能体强化学习中，集中训练与分布式执行 (CTDE) 是一种有前景的多无人机避障方法，其中的关键挑战是有效地学习分布式策略，这些策略能够合作最大化全局奖励。我们提出了一个新的多智能体评价-行动学习方案，称为 MACA，用于无人机群避障。MACA 使用一个集中评价器来最大化考虑安全性和能效的折现全局奖励，以及每个无人机上的一个行动器来寻找避免碰撞的分布式策略。为了解决 CTDE 中的信用分配问题，我们设计了一个反事实基线，该基线消除了智能体的状态和动作，使得能够在联合观察-动作空间中评估智能体的重要性。为了训练和评估 MACA，我们设计了自己的仿真环境 MACAEnv，以逼真地模拟无人机群的实际行为。仿真结果显示，MACA 比两种最先进的多智能体强化学习算法的平均奖励高出 16% ，并将失败率降低了 90% ，在所有测试场景中，响应时间缩短了超过 99% 。

I. INTRODUCTION

I. 引言

UAV swarms have become popular in many practical scenarios such as surveillance, transportation, agriculture, search and rescue in hardly-accessible areas and so on. A UAV swarm is a group of UAVs collaborating with each other to accomplish a common mission. Each member in a UAV swarm needs to avoid collisions with not only the obstacles (e.g. birds and adversary UAVs) but also the neighboring swarm members. If each member selects actions independently, more collisions may occur as the action chosen by one member will affect other members. Hence, cooperative path planning is needed for collision avoidance in a UAV swarm, which still remains a critical challenge.

无人机群在许多实际场景中变得流行，例如监控、运输、农业、难以到达区域的搜索和救援等。无人机群是一组相互协作的无人机，共同完成任务。无人机群中的每个成员都需要避免与障碍物 (例如鸟类和敌方无人机) 以及邻近群成员发生碰撞。如果每个成员独立选择行动，由于一个成员选择的行为会影响其他成员，可能会发生更多的碰撞。因此，无人机群中需要进行合作路径规划以避免碰撞，这仍然是一个关键挑战。

Conventional swarm planning methods suffer from drawbacks including high computational complexity, long reaction time and the need of global environment information to generate collision-free paths. For example, centralized approaches [1], [2] use heuristic algorithms such as A* to select optimal collision-free paths by considering all possible way points in the environment at each planning step. Except the need of knowledge on global environment, the computational complexity grows significantly with the increase on environment size. Decentralized methods such as E^2 Coop [3] allows each swarm member to plan its energy-efficient and collision-free trajectories using Particle Swarm Optimization (PSO) combined with Artificial Potential Field (APF), but rely on highly reliable UAV-to-UAV communication for cooperative collision avoidance. Considering the constraints on computing resources and battery lifetime for UAVs

as well as the complexity of air traffic, low-complexity collision avoidance methods with short reaction time and high energy-efficiency is more desirable for UAV swarms.

传统的群体规划方法存在一些缺点，包括计算复杂度高、反应时间长以及需要全局环境信息来生成无碰撞路径。例如，集中式方法 [1]、[2] 使用启发式算法如 A* 来选择无碰撞的最优路径，通过在每个规划步骤考虑环境中所有可能的路径点。除了需要全局环境知识外，计算复杂度随着环境规模的增加而显著增长。去中心化方法如 E^2 Coop [3] 允许每个群体成员使用粒子群优化 (PSO) 结合人工势场 (APF) 规划其节能且无碰撞的轨迹，但依赖于高度可靠的无人机间通信来实现合作避碰。考虑到无人机的计算资源和电池寿命限制以及空域复杂性，对于无人机群来说，具有短反应时间和高能效的低复杂度避碰方法更为理想。

Multi-Agent Reinforcement Learning (MARL) is a promising method for multi-UAV collision avoidance by modeling the problem as a decentralized partially observable Markov decision process (Dec-POMDP), considering UAVs in a swarm as agents in RL. Especially, the Centralized Training with Decentralized Execution (CTDE) paradigm in MARL allows to train multiple agents in a centralized manner using local and global information that cannot be acquired in execution, while in execution, agents make decisions based on only local observations in a decentralized way. This significantly shifts the complexity to training and makes execution extremely lightweight. In order to train multiple agents to achieve the best team performance, it is important to maximize a global reward in the centralized training. However, when agents learn from a global reward independently, the lazy agent problem arises, where a bad action made by one agent may be rewarded due to the good actions performed by others agents. Credit assignment, which attributes the global reward according to the contribution of each agent, is of crucial importance in learning effective decentralized policies. Value factorization based methods such as VDN [4], QMix [5], and QTRAN [6] train a local critic for each agent and factorize the local critic values into a global value. However, this requires to explicitly define the relation between the global and local values, which is nontrivial. Another class of methods such as COMA [7] and Shapley [8] trains a centralized critic for all agents. An advantage function is explicitly derived for each agent to estimate its contribution from the centralized critic using difference rewards. However, COMA is designed for discrete action space. Although it can be extended to continuous action space using Gaussian approximation, the performance is poor when the sampling granularity is low. As a result, the actors fail to learn optimal policies and the trajectories are neither energy efficient nor safe. When the sampling granularity is high, the network structure becomes complex and the computational complexity increases. Shapley also has a high computational complexity since its advantage function is computed over all possible agent combinations.

多智能体强化学习 (MARL) 是一种有前景的多无人机避障方法，通过将问题建模为分布式部分可观测马尔可夫决策过程 (Dec-POMDP)，将无人机群中的无人机视为强化学习中的智能体。特别是，MARL 中的集中训练与分布式执行 (CTDE) 范式允许使用无法在执行中获取的局部和全局信息以集中方式训练多个智能体，而在执行过程中，智能体仅基于局部观测以分布式方式进行决策。这显著地将复杂性转移到了训练上，并使得执行变得极其轻量级。为了训练多个智能体以达到最佳的团队表现，重要的是在集中训练中最大化全局奖励。然而，当智能体独立地从全局奖励中学习时，会出现懒惰智能体问题，其中一个智能体的不良行为可能因为其他智能体的良好行为而得到奖励。信用分配，即将全局奖励根据每个智能体的贡献进行归因，在学习有效的分布式策略中至关重要。基于价值分解的方法，如 VDN [4]、QMix [5] 和 QTRAN [6]，为每个智能体训练一个局部评估器并将局部评估器价值分解为全局价值。然而，这需要显式定义全局和局部价值之间的关系，这是非平凡的。另一类方法，如 COMA [7] 和 Shapley [8]，为所有智能体训练一个集中评估器。为每个智能体显式导出一个优势函数，以从集中评估器中使用差异奖励估计其贡献。然而，COMA 是为离散动作空间设计的。尽管可以使用高斯近似将其扩展到连续动作空间，但在采样粒度较低时性能较差。因此，执行器无法学习到最优策略，且轨迹既不节能也不安全。当采样粒度较高时，网络结构变得复杂，计算复杂度增加。Shapley 的计算复杂度也较高，因为其优势函数是在所有可能的智能体组合上计算的。

In this paper, we present MACA, a Multi-Agent reinforcement learning based scheme with Counterfactual credit Assignment for collision avoidance in UAV swarms. MACA consists of a centralized critic network and a local actor network for each swarm member. The critic and actor networks are trained to optimize a global reward that considers both safety and energy efficiency. Unlike COMA, MACA is designed for continuous action space and thus need neither sampling nor approximation. Compared with Shapley, its advantage function can be computed by a simple forward pass of the critic with low complexity. To the best of our knowledge, MACA is the first to use multi-actor centralised critic reinforcement

The authors are with Department of Computer Science, Univer- {shuangyao, haibo, hzy}@cs.otago.ac.nz
作者来自奥塔哥大学计算机科学系，邮箱为 {shuangyao, haibo, hzy}@cs.otago.ac.nz

learning for multi-UAV collision avoidance. The main contributions can be summarized as:

在本文中，我们提出了 MACA，一种基于多智能体强化学习的方案，采用反事实信用分配来避免无人机群中的碰撞。MACA 包括一个集中式评判网络和每个群体成员的局部执行网络。评判网络和执行网络被训练以优化一个考虑安全性和能效的全局奖励。与 COMA 不同，MACA 专为连续动作空间设计，因此无需采样或近似。与 Shapley 相比，其优势函数可以通过评判网络的简单前向传播以低复杂度计算得出。据我们所知，MACA 是首个将多执行者集中评判强化学习应用于多无人机避碰问题的方案。主要贡献可以概括为:

- We formulate the path planning problem for collision avoidance in a UAV swarm as a decentralized partially observable Markov decision process (Dec-POMDP). A new reward function is designed by considering both safety and energy efficiency in collision avoidance.
- 我们将无人机群中的避碰路径规划问题表述为一个分布式部分可观测马尔可夫决策过程 (Dec-POMDP)。通过考虑避碰中的安全性和能效，设计了一种新的奖励函数。
- We design a multi-actor centralized critic learning scheme for UAV swarm collision avoidance. A new advantage function with low complexity is designed. We theoretically prove the convergence of our learning scheme.
- 我们为无人机群避碰设计了一个多执行者集中评判学习方案。设计了一种新的低复杂度优势函数。我们从理论上证明了我们的学习方案的收敛性。
- We also design an emergency avoidance scheme at the control level to complement the actor model to further boost the success rate of collision avoidance.
- 我们还在控制级别设计了一个紧急避碰方案，以补充执行模型，进一步提高避碰成功率。
- We design a MARL environment named MACAEnv to train and evaluate MACA. Extensive experiments are conducted to demonstrate the effectiveness and efficiency of our scheme.
- 我们设计了一个名为 MACAEnv 的 MARL 环境来训练和评估 MACA。进行了大量实验来证明我们方案的有效性和效率。

II. RELATED WORK

II. 相关工作

A. Collision Avoidance For UAV Swarms

A. 无人机群的碰撞避免

Intensive research has been conducted on collision avoidance for UAV swarms. Conventional methods include Velocity Obstacles (VO) [9], Artificial Potential Field (APF) [10], Particle Swarm Optimization (PSO) [11] and hybrid methods [3]. VO based methods require reliable communications among UAVs. The trajectories planned using VO are usually Zig-Zag, as the UAVs change velocities frequently. Hence, the trajectories are not energy efficient. APF based methods didn't address the problem of coordination among UAVs. Repulsive forces added between UAVs to avoid UAV-to-UAV collisions may generate Zig-Zag trajectories like in VO. Hybrid methods combining APF and PSO require large online response time, as the cost function in PSO is usually complex. Decentralized Model Predictive Control (MPC) has been utilized to solve the problem of UAVs collision avoidance [12]. However, MPC follows a leader-follower structure and highly depends on UAV-to-UAV communications. The whole swarm will be in danger if the leader dies or the communication network is interfered.

对无人机群避障进行了深入研究。传统方法包括速度障碍 (VO)[9]、人工势场 (APF)[10]、粒子群优化 (PSO)[11] 以及混合方法 [3]。基于 VO 的方法需要无人机之间可靠的通信。使用 VO 规划出的轨迹通常是曲折的，因为无人机需要频繁改变速度。因此，这些轨迹在能源效率方面不高。基于 APF 的方法没有解决无人机之间的协调问题。在无人机之间添加的排斥力以避免无人机相撞可能会产生类似于 VO 的曲折轨迹。结合 APF 和 PSO 的混合方法由于 PSO 中的成本函数通常很复杂，需要较长的在线响应时间。分布式模型预测控制 (MPC) 已被用于解决无人机避障问题 [12]。然而，MPC 遵循领导者-跟随者结构，并且高度依赖于无人机之间的通信。如果领导者失效或通信网络受到干扰，整个群体将处于危险之中。

Independent reinforcement learning has been proposed to better solve the problem of collision avoidance for UAV swarms. Fully decentralized algorithms [13], [14], [15] suffer non-stationary problems and require UAV-to-UAV or UAV-to-ground communications. Therefore, CTDE based algorithms were designed to address the drawbacks of fully centralized algorithms. MADDPG [16] is a well known MARL algorithm based on CTDE with individual reward. However, MADDPG cannot guarantee the best team performance as agents in MADDPG are trained by individual rewards rather than a global reward, and thus it is not suitable to be used for UAV collision avoidance.

已经提出了独立强化学习，以更好地解决无人机群避障问题。完全分布式算法 [13]、[14]、[15] 存在非平稳问题，并且需要无人机之间或无人机与地面之间的通信。因此，基于 CTDE 的算法被设计出来，以解决完全集中式算法的缺点。MADDPG[16] 是基于 CTDE 的知名多智能体强化学习 (MARL) 算法，具有个体奖励。然而，由于 MADDPG 中的智能体是通过个体奖励而不是全局奖励进行训练的，因此它不能保证最佳团队表现，不适合用于无人机避障。

B. Credit Assignment in MARL

B. 多智能体强化学习中的信用分配

In order to train agents to maximize a global reward, it is important to solve the credit assignment problem. Existing credit assignment methods in MARL can be divided in two classes: implicit assignment and explicit assignment.

为了训练智能体以最大化全局奖励，解决信用分配问题非常重要。现有 MARL 中的信用分配方法可以分为两类: 隐式分配和显式分配。

Examples of implicit credit assignment include VDN [4], QMix [5] and QTRAN [6]. VDN solves the problem of credit assignment by implicitly decomposing the central Q value Q_{tot} to local utility Q_i values additively. QMix factorizes the central critic Q_{tot} monotonically with an additional mixer network that decides weights and bias for Q_i . QTRAN trains a state-value network in addition to VDN to generalize the representation of Q_{tot} . These algorithms cover a wide range of representations of Q_{tot} , but they usually have high computational complexity and require large network size.

隐式信用分配的例子包括 VDN [4]、QMix [5] 和 QTRAN [6]。VDN 通过隐式地将中心 Q 价值 Q_{tot} 分解为局部效用 Q_i 值的加和来解决信用分配问题。QMix 通过一个额外的混合网络决定 Q_i 的权重和偏置，单调地分解中心评估器 Q_{tot} 。QTRAN 除了 VDN 之外，还训练一个状态价值网络以泛化 Q_{tot} 的表示。这些算法涵盖了 Q_{tot} 的广泛表示，但它们通常具有高计算复杂度且需要大型网络规模。

COMA [7] is a representative of MARL algorithms with explicit credit assignment, built on the idea of difference rewards [17]. COMA instead utilizes a centralized critic network to approximate an advantage function based on a counterfactual baseline, which is acquired by averaging over the Q values of all possible actions of an agent except the taken action. This requires the centralized critic network to have at least $|\mathcal{A}|$ output heads, where $\mathcal{A}$ is the set of discrete actions of an agent. Therefore, the calculation of baseline can only be performed well on discrete action spaces.

COMA [7] 是基于差异奖励 [17] 思想的显式信用分配多智能体学习算法的代表。COMA 转而使用集中式评估网络来基于反事实基线近似优势函数，该基线是通过除采取的行动之外的所有可能行动的 Q 值求平均得到的。这要求集中式评估网络至少有 $|\mathcal{A}|$ 个输出头，其中 $\mathcal{A}$ 是智能体的离散动作集。因此，基线的计算只能在离散动作空间上良好执行。

Inspired by COMA, a similar idea has been proposed by [8], where an advantage function for an agent is approximated with the Shapley value of game theory, which is a weighted sum of Q values over all possible agent collations including the ego agent. The computational complexity of the Shapley value is $O(N!)$, where N is the number of agents. In comparison with COMA and Shapley, our scheme MACA has much simpler network structures and lower computational complexity, and is also applicable to continuous action space.

受 COMA 启发，[8] 提出了一个类似的想法，其中智能体的优势函数用博弈论中的 Shapley 值来近似，这是包括自我智能体在内的所有可能智能体联合的 Q 值的加权和。Shapley 值的计算复杂度为 $O(N!)$ ，其中 N 是智能体的数量。与 COMA 和 Shapley 相比，我们的方案 MACA 具有更简单的网络结构，更低的计算复杂度，并且也适用于连续动作空间。

III. SYSTEM MODEL AND FRAMEWORK

III. 系统模型与框架

A. System Model

A. 系统模型

Each UAV is equipped with a Lidar sensor to detect the positions and velocities of obstacles and its neighboring UAVs. For example, Lidar sensors such as SICK LD-MRS [18] have a horizontal aperture angle of 110° , a working range of 150 m and weights less than 1Kg .

每个无人机都配备了一个激光雷达传感器，用于检测障碍物及其邻近无人机的位置和速度。例如，像 SICK LD-MRS [18] 这样的激光雷达传感器具有 110° 的水平视场角，150 m 的工作范围，重量小于 1Kg 。

A GPS sensor is also required for self-localization. We assume large static obstacles such as buildings have been bypassed in off-line mission planning. So we focus on avoiding collisions with small dynamic obstacles such as adversary UAVs and birds. Hence, it's reasonable to model obstacles as moving points. However, our method can be extended to deal with large obstacles with any shapes by treating them as discrete points along obstacle's boundary.

还需要一个 GPS 传感器来进行自我定位。我们假设在离线任务规划中已经绕过了大型静态障碍物，如建筑物。因此，我们专注于避免与小型动态障碍物，如敌对无人机和鸟类相撞。因此，将障碍物建模为移动点是合理的。然而，我们的方法可以通过将大型障碍物视为障碍物边界上的离散点来扩展以处理任何形状的大型障碍物。

In most applications of UAV swarms, UAVs fly at a constant speed along a pre-planned path. When obstacles are detected, UAVs take actions cooperatively to avoid collisions. After collision avoidance, UAVs come back to their preplanned paths to continue the mission. To ensure safety, the distance between swarm members and obstacles should not be smaller than a safeguard distance d^{obs} and the distance between swarm members should not be smaller than a safeguard distance d^{v2v} . Since both safety and energy efficiency are of critical importance for long-haul UAV applications, our optimization goal is to generate paths for swarm members to avoid collisions with high energy efficiency.

在大多数无人机群应用中，无人机沿着预先规划好的路径以恒定速度飞行。当检测到障碍物时，无人机合作采取行动以避免碰撞。避障后，无人机返回到它们预先规划好的路径以继续任务。为确保安全，群成员与障碍物之间的距离不应小于安全距离 d^{obs} ，群成员之间的距离也不应小于安全距离 d^{v2v} 。由于在长途无人机应用中，安全和能量效率都至关重要，我们的优化目标是生成群成员的路径，以高能量效率避免碰撞。

B. Framework of MACA

B. MACA 框架

The key idea of MACA is to exploit Multi-Agent Reinforcement Learning (MARL) under the CTDE paradigm to achieve low-complexity and energy-efficient collision avoidance. As illustrated in Fig. 1, the MARL method adopts an actor-critic model with a centralized critic. In the centralized training phase, the central critic model is trained to approximate the cumulative global reward that measures both safety and energy efficiency. We design an agent-specific advantage function to approximate the contribution of each agent by marginalizing out both the agent's observation and action while keeping other agents unchanged. Actor models are then trained to learn policies that maximize the corresponding advantage functions. In the decentralized execution phase, each member in the swarm chooses the action to avoid collision using its actor model based on only its local observation. Although all swarm members perform collision avoidance in a decentralized way based on only local observations, the actor models are trained to act cooperatively in a centralized way. Hence, the actions selected by the swarm members in execution are still cooperative with each other.

MACA 的关键思想是在 CTDE 范式下利用多智能体强化学习 (MARL) 实现低复杂度和节能的避障。如图 1 所示，MARL 方法采用带有集中式评价器的演员-评价器模型。在集中式训练阶段，中心评价器模型被训练以近似累计全局奖励，该奖励衡量了安全性和能效。我们设计了一个特定于智能体的优势函数，通过边际化出每个智能体的观测和动作，同时保持其他智能体不变，来近似每个智能体的贡献。然后训练演员模型来学习最大化相应优势函数的策略。在去中心化执行阶段，群体中的每个成员仅基于其本地

观测使用其演员模型选择避障动作。尽管所有群体成员仅基于本地观测以去中心化的方式进行避障，但演员模型是以集中化的方式进行合作训练的。因此，执行中群体成员选择的行为仍然是相互合作的。

Safety is of critical importance to UAV applications, whereas the actor models centrally trained with global information cannot always guarantee to choose the correct actions to avoid collisions. An Emergency Avoidance Scheme (EAS) is designed to complement the actor models for further boosting safety. If the action selected by the actor is infeasible, EAS will choose a feasible action close to the action selected by the actor model.

安全性对于无人机应用至关重要，而使用全局信息集中训练的演员模型并不能总是保证选择正确的动作来避免碰撞。因此，设计了一个紧急避障方案 (EAS) 来补充演员模型，以进一步提高安全性。如果演员选择的动作不可行，EAS 将选择一个接近演员模型所选动作的可行动作。

IV. CENTRALIZED TRAINING

IV. 集中式训练

A. Dec-POMDP Model

A. 分布式部分可观测马尔可夫决策过程 (Dec-POMDP) 模型

The objective of MACA is to generate paths for UAVs in a swarm to cooperatively avoid collisions while minimizing energy consumption. Such a problem can be modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) defined by a tuple(S, O, A, R, P):

MACA 的目标是生成无人机群体的路径，以合作避障同时最小化能耗。这样一个问题可以被建模为一个分布式部分可观测马尔可夫决策过程 (Dec-POMDP)，由一个元组 (S, O, A, R, P) 定义:

1) State space $\mathcal{S}$ and observation space $\mathcal{O}$: the state at time t , denoted by $s_t \in \mathcal{S}$, is a true state of the global environment defined by the positions and velocities of all swarm members and the obstacles detected at time t . Due to limited sensing range, each UAV may only observe a part of the true environment. The observation of the i^{th} UAV at time t , denoted by $o_t^i \in \mathcal{O}$, is represented by an information list that consists of both internal observations and external observations. The internal observations include the UAV's current position, velocity and pre-planned velocity. The external observations include the positions and velocities of other swarm members and obstacles detected by the Lidar sensor. Based on these definitions, we have $s_t \approx o_t = \cup_{i=1}^{N} o_i$ where N is the number of UAVs in the swarm.

1) 状态空间 $\mathcal{S}$ 和观测空间 $\mathcal{O}$: 在时间 t 的状态，表示为 $s_t \in \mathcal{S}$ ，是全局环境的真实状态，由所有群体成员和 t 时刻检测到的障碍物的位置和速度定义。由于感知范围有限，每个无人机可能只能观察到真实环境的一部分。i^{th} 无人机在 t 时刻的观测，表示为 $o_t^i \in \mathcal{O}$ ，由内部观测和外部观测组成的信息列表。内部观测包括无人机的当前位置、速度和预计划速度。外部观测包括其他群体成员和由激光雷达传感器检测到的障碍物的位置和速度。基于这些定义，我们有 $s_t \approx o_t = \cup_{i=1}^{N} o_i$ ，其中 N 是群体中无人机的数量。

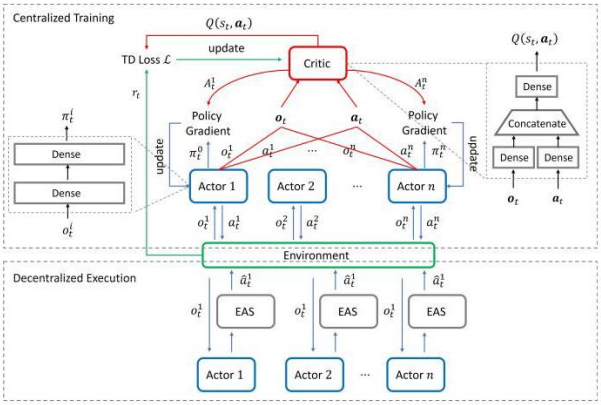


Fig. 1: The framework of MACA.

图 1:MACA 的框架。

2) Action Space $\mathcal{A}$: To improve energy efficiency, we confine collision avoidance in a two-dimensional space as altitude flight can consume much more energy than level flight [3]. The control of a UAV can be simplified to the control of Yaw angle when flying with constant speed in an airspace where the air speed can be ignored [3]. However, our scheme can be easily extended to three-dimensional space by setting the control action space to { Yaw, Pitch, Roll } angles. In training, we design the action taken by the i^{th} UAV to be $a_i \in [-1, 1]$, corresponding to an Yaw angle change of $[-45°, 45°]$, in order to stabilize training.

2) 动作空间 $\mathcal{A}$: 为了提高能量效率，我们将避障限制在二维空间内，因为与平飞相比，高度飞行会消耗更多的能量 [3]。当在忽略空气速度的空域中以恒定速度飞行时，无人机的控制可以简化为偏航角的控制 [3]。然而，我们的方案可以通过将控制动作空间设置为 { 偏航角、俯仰角、滚转角 } 来轻松扩展到三维空间。在训练过程中，我们设计 i^{th} 无人机的动作是 $a_i \in [-1, 1]$ ，对应于偏航角变化 $[-45°, 45°]$ ，以稳定训练。

3) Reward Function $\mathcal{R}$: To avoid collisions and minimize energy consumption, the reward function is designed to have an internal reward ${R_t}^{\text{Int}}$ that measures energy efficiency and an external reward R_t^{Ext} that reflects safety.

3) 奖励函数 $\mathcal{R}$: 为了避免碰撞并最小化能量消耗，奖励函数被设计为具有衡量能量效率的内部奖励 ${R_t}^{\text{Int}}$ 和反映安全的外部奖励 R_t^{Ext} 。

$$R_t = R_t^{Int} + R_t^{Ext}$$

$$R_t^{Int} = \sum_{i=1}^{N} \left\langle \frac{\mathbf{v}_t^i}{|\mathbf{v}_t^i|}, \frac{\overline{\mathbf{v}}_t^i}{|\overline{\mathbf{v}}_t^i|} \right\rangle - |a_i| - \frac{\left|\mathbf{x}_t^i, \overline{\mathbf{x}}_t^i\right|}{D^{\max}}, \tag{1}$$

$$R_{Ext} = \begin{cases} 0 & \text{if Collision } = \text{ False} \\ -C & \text{if Collision } = \text{ True} \end{cases}$$

where $\mathbf{v}_t^i$ and $\overline{\mathbf{v}}_t^i$ are the current and pre-planned speed of the i^{th} UAV respectively, and $\langle \cdot \rangle$ is inner product. $\mathbf{x}_t^i$ is the current position and $\overline{\mathbf{x}}_t^i$ is the pre-planned position at time t . So, $|\mathbf{x}_i, \overline{\mathbf{x}}_i|$ measures the distance between the current position and the pre-planned position. $D^{\max}$ is an environment specific parameter defining the maximum distance a UAV can deviate from its pre-planned path. The first term in ${R_t}^{\text{Int}}$ guides the UAVs to recover their original speeds after avoidance. The second term rewards flying along smooth trajectory since the smoother the trajectories the less energy required [3]. The third term aims to drag the UAVs back to their pre-planned paths after avoidance. When either the distance between a UAV and its neighbors falls below the safeguard distance d^{v2v} or the distance between a UAV and an obstacle falls below the safeguard distance d^{obs} , it is assumed collision occurs. In R_{Ext}, C is a model hyper-parameter to penalize collisions.

其中 $\mathbf{v}_t^i$ 和 $\overline{\mathbf{v}}_t^i$ 分别是 i^{th} 无人机的当前速度和预先计划的速度，$\langle \cdot \rangle$ 是内积。$\mathbf{x}_t^i$ 是当前位置，$\overline{\mathbf{x}}_t^i$ 是在时间 t 的预先计划位置。因此，$|\mathbf{x}_i, \overline{\mathbf{x}}_i|$ 衡量了当前位置与预先计划位置之间的距离。$D^{\max}$ 是一个特定于环境的参数，定义了无人机可以偏离其预先计划路径的最大距离。在 ${R_t}^{\text{Int}}$ 中的第一项指导无人机在避障后恢复其原始速度。第二项奖励沿平滑轨迹飞行，因为轨迹越平滑，所需的能量越少 [3]。第三项旨在将无人机在避障后拖回其预先计划的路径。当无人机与其邻居的距离低于安全距离 d^{v2v} 或者无人机与障碍物的距离低于安全距离 d^{obs} 时，假设发生碰撞。在 R_{Ext}, C 中是一个模型超参数，用于对碰撞进行惩罚。

In each training episode, the swarm gets a global reward from the environment as a result of the actions performed by all swarm members. All UAVs in the swarm select actions to maximize the discounted long-term reward $\mathcal{R} = \sum_{l=0}^{\infty} \gamma^l R_{t+l}$, where $\gamma \in [0, 1)$ is a discounted factor.

在每个训练阶段，群体由于所有群体成员执行的动作而从环境中获得全局奖励。群体中的所有无人机选择动作以最大化折现后的长期奖励 $\mathcal{R} = \sum_{l=0}^{\infty} \gamma^l R_{t+l}$ ，其中 $\gamma \in [0, 1)$ 是折现因子。

B. Architecture of multi-actor centralized critic

B. 多演员集中式评判者的架构

The centralized critic processes inputs using a dense layer with 64 neurons, of which 32 neurons are used for joint observations and the others are used for joint actions. The observations and actions are concatenated as inputs for two dense layers with 256 neurons followed by ReLU activation functions. Finally, a dense output layer produces $Q(s, \mathbf{a})$. The actor network has a simple architecture consisting

of two dense layers with 256 hidden units, followed by ReLU activation functions. Finally, the output layer gives an action a_i using a tanh activation function.

集中式评判器使用具有 64 个神经元的密集层处理输入，其中 32 个神经元用于联合观察，其余用于联合行动。观察和行动被连接作为两个具有 256 个神经元的密集层的输入，随后是 ReLU 激活函数。最终，一个密集输出层产生 $Q(s,\mathbf{a})$ 。演员网络具有简单的架构，包括两个具有 256 个隐藏单元的密集层，随后是 ReLU 激活函数。最后，输出层使用 tanh 激活函数给出一个行动 a_i 。

The centralized critic $f^c(s,\mathbf{a},\omega)$ is trained to approximate the action value function $Q(s,\mathbf{a}) = \mathbb{E}_{s_t,\mathbf{a}_t}[\mathcal{R}_t \mid s_t,\mathbf{a}_t]$ for the joint action $\mathbf{a}$ and global state s , where $\mathcal{R}_t = \sum_{l=0}^{\infty}\gamma^l r_{t+l}$ is the discounted cumulative reward. Specifically, the critic parameters ω are updated by minibatch gradient descent that minimizes the following loss:

集中式评判器 $f^c(s,\mathbf{a},\omega)$ 被训练来近似联合行动 $\mathbf{a}$ 和全局状态 s 的动作价值函数 $Q(s,\mathbf{a}) = \mathbb{E}_{s_t,\mathbf{a}_t}[\mathcal{R}_t \mid s_t,\mathbf{a}_t]$ ，其中 $\mathcal{R}_t = \sum_{l=0}^{\infty}\gamma^l r_{t+l}$ 是折现累积奖励。具体来说，评判器参数 ω 通过最小化以下损失的小批量梯度下降来更新:

$$\mathcal{L}_t^c = \left(y_t - f_t^c(s,\mathbf{a},\omega)\right)^2, \tag{2}$$

$$y_t = r_t + \gamma \cdot f_t^c(s,\mathbf{a},\bar{\omega}),$$

where $\bar{\omega}$ are parameters of a target network which are periodically updated by ω [19]. The target network has the same structure with the centralized critic, but has its own parameter set. The target network is used to cut bootstrap in calculating $\mathcal{L}_t^c$ to stabilize the training process.

其中 $\bar{\omega}$ 是目标网络的参数，该参数通过 ω [19] 定期更新。目标网络具有与集中式评判器相同的结构，但是拥有自己的参数集。目标网络用于在计算 $\mathcal{L}_t^c$ 时切断引导，以稳定训练过程。

Each actor i is then trained following a policy gradient:

然后，每个演员 i 都遵循策略梯度进行训练:

$$g_i = \mathbb{E}_\pi\left[\nabla_\theta \log \pi_i(s_i,\theta) A_i(s,\mathbf{a})\right] \tag{3}$$

where $A_i(s,\mathbf{a})$ is the advantage function for actor i . Unlike COMA and Shapley in which their advantage functions marginalize only the action a_i , our advantage function $A_i(s,\mathbf{a})$ marginalizes both o_i and a_i as follows:

其中 $A_i(s,\mathbf{a})$ 是演员 i 的优势函数。与 COMA 和 Shapley 不同，它们的优势函数只边际化行动 a_i ，我们的优势函数 $A_i(s,\mathbf{a})$ 则边际化 o_i 和 a_i ，如下所示:

$$A_i(s,\mathbf{a}) = Q(s,\mathbf{a}) - Q(s-o_i,\mathbf{a}-a_i), \tag{4}$$

where $Q(s-o_i,\mathbf{a}-a_i)$ is computed using the critic by replacing both o_i and a_i with null, as if actor i observes nothing and takes no action. In COMA and Shapley, the counterfactual baseline is calculated by replacing the action while keeping the others unchanged. This can be seen as evaluating the contribution of an agent's action given the agent's observation in a one dimensional space. While in our scheme, the counterfactual baseline is calculated by replacing both an actor's observation and action. This can be seen as evaluating the importance of an agent (an observation-action pair) in a two dimensional space. In comparison with COMA and Shapley, our advantage function is also much simpler as it can be computed by a single pass of the critic, whereas the time complexity for the advantage function in COMA is $O(|\mathcal{A}|)$ as its baseline is summed over all possible actions of an actor, and the time complexity for the advantage function in Shapley is $O(N!)$ as its baseline is summed over all possible actor combinations.

其中 $Q(s-o_i,\mathbf{a}-a_i)$ 是通过评判者计算得出的，通过将 o_i 和 a_i 都替换为空值，就像执行者 i 没有观察到任何东西并且没有采取任何行动一样。在 COMA 和 Shapley 方法中，反事实基线是通过替换动作而保持其他不变来计算的。这可以看作是在一维空间中评估给定执行者观察到的动作的贡献。而在我们的方案中，反事实基线是通过替换执行者的观察和动作来计算的。这可以看作是在二维空间中评估一个智能体 (观察-动作对) 的重要性。与 COMA 和 Shapley 相比，我们的优势函数也更简单，因为它可以通过评判者的单次遍历来计算，而 COMA 中的优势函数的时间复杂度为 $O(|\mathcal{A}|)$ ，因为其基线是对执行者所有可能的动作求和，Shapley 中的优势函数的时间复杂度为 $O(N!)$ ，因为其基线是对所有可能的执行者组合求和。

Lemma 1: If the global Q-value is a linear combination of all individual contributions, $A_i(s,\mathbf{a})$ is the contribution made by actor i at state s with action a_i when the centralized critic converges to represent the global Q-value function.

引理 1: 如果全局 Q 值是所有个体贡献的线性组合，$A_i(s, \mathbf{a})$ 是当集中式评判者收敛于表示全局 Q 值函数时，执行者 i 在状态 s 下采取动作 a_i 的贡献。

Proof: We use $Q_i(s, o_i, a_i)$ to denote the utility value of actor i with state s , observation o_i and action a_i . Note that it is called the utility value rather than Q value because this value does not follow Bellman equation. Let $m(\cdot)$ be the function mapping Q values of individual actors to the global Q value, that is,

证明: 我们用 $Q_i(s, o_i, a_i)$ 来表示执行者 i 在状态 s 、观察 o_i 和动作 a_i 下的效用值。注意，这里称之为效用值而不是 Q 值，因为此值不遵循贝尔曼方程。令 $m(\cdot)$ 为将个体执行者的 Q 值映射到全局 Q 值的函数，即，

$$Q(s, \mathbf{a}) \rightarrow m\left(Q_0(s, o_0, a_0), Q_1(s, o_1, a_1), \cdots, Q_N(s, o_N, a_N)\right)$$

Since the global Q-value is a linear combination of all individual contributions, we have

由于全局 Q 值是所有个体贡献的线性组合，我们有

$$Q(s, \mathbf{a}) = \sum_{i=1}^{N} Q_i(s, o_i, a_i) \tag{5}$$

Based on the advantage function defined in Eq. 4,

基于在公式 4 中定义的优势函数，

$$A_i(s, \mathbf{a}) = Q(s, \mathbf{a}) - Q(s - o_i, \mathbf{a} - a_i)$$

$$= m(\cdots, Q_{i-1}, Q_i, Q_{i+1}, \cdots) - m(\cdots, Q_{i-1}, Q_{i+1}, \cdots)$$

$$= \sum_{j=1}^{N} Q_j(s, o_j, a_j) - \sum_{j=1}^{i-1} Q_j(s, o_j, a_j) - \sum_{j=i+1}^{N} Q_j(s, o_j, a_j)$$

$$= Q_i(s, o_i, a_i)$$

According to the reward function defined in Eq. 1), the global reward is a linear combination of all local rewards. In the following lemma, we show that the baseline used in Eq. (4) does not affect the convergence of the centralized critic.

根据 (1) 式中定义的奖励函数，全局奖励是所有局部奖励的线性组合。在以下引理中，我们证明了 (4) 式中使用的基线不会影响集中式评判器的收敛性。

Lemma 2: For a critic following our policy gradient at step t :

引理 2: 对于在步骤 t 遵循我们的策略梯度的评判器:

$$g_t = \mathbb{E}_{\pi}\left[\sum_i \nabla_\theta \log \pi_i(s_i, a_i) A_i(s, \mathbf{a})\right], \tag{6}$$

$$A_i(s, \mathbf{a}) = Q(s, \mathbf{a}) - Q(s - o_i, \mathbf{a} - a_i),$$

the state-action dependent baseline $Q(s - o_i, \mathbf{a} - a_i)$ doesn' t introduce any bias to the critic, and hence does not affect the convergence of the policy gradient algorithm.

状态-动作依赖的基线 $Q(s - o_i, \mathbf{a} - a_i)$ 不会给评判器引入任何偏差，因此不会影响策略梯度算法的收敛性。

Proof: Let $\pi(s)$ be the joint policy which is a product of the policy for all actors as follows: $\pi(s) = \prod_i \pi_i(s_i)$. Then the proof for this lemma can follow the same logic for the convergence proof of single agent actor-critic algorithm [20]. Let $d^{\pi}(s)$ be the discounted state distribution defined as follows: $d^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s \mid s_0, \pi)$ [20]. Then the expectation on baseline is:

证明: 设 $\pi(s)$ 为联合策略，它是所有演员策略的乘积，如下所示: $\pi(s) = \prod_i \pi_i(s_i)$ 。那么这个引理的证明可以遵循单一智能体演员-评判器算法收敛证明的相同逻辑 [20]。设 $d^{\pi}(s)$ 为折扣状态分布，定义为: $d^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s \mid s_0, \pi)$ [20]。那么基线的期望是:

$$
\begin{aligned}
g_b &= -\mathbb{E}_\pi \left[\sum_i \nabla_\theta \log \pi_i \left(s_i, a_i\right) Q\left(s - o_i, \mathbf{a} - a_i\right) \right] \\
&= -\sum_s d^\pi(s) \sum_i \sum_{s^{-i}} \sum_{\mathbf{a}^{-i}} \pi\left(s - o_i, \mathbf{a} - a_i\right). \\
&\sum_{s^i} \sum_{a^i} \pi_i \left(s_i, a_i\right) \nabla_\theta \log \pi_i \left(s_i, a_i\right) Q\left(s - o_i, \mathbf{a} - a_i\right) \\
&= -\sum_s d^\pi(s) \sum_i \sum_{s^{-i}} \sum_{\mathbf{a}^{-i}} \pi\left(s - o_i, \mathbf{a} - a_i\right). \\
&\sum_{s^i} \sum_{a^i} \nabla_\theta \pi_i \left(s_i, a_i\right) Q\left(s - o_i, \mathbf{a} - a_i\right) \\
&= -\sum_s d^\pi(s) \sum_i \sum_{s^{-i}} \sum_{\mathbf{a}^{-i}} \pi\left(s^{-i}, \mathbf{a}^{-i}\right) Q\left(s - o_i, \mathbf{a} - a_i\right) \nabla_\theta \\
&= 0 .
\end{aligned}
$$

As shown above, the baseline doesn't introduce bias although it is state-action dependent. Hence, the baseline doesn't affect the convergence of the algorithm.

如上所示，尽管基线是状态-动作依赖的，但它不会引入偏差。因此，基线不会影响算法的收敛性。

V. DECENTRALIZED EXECUTION

V. 分布式执行

Only actor models are used in execution. Actor i takes only its local observation o_i as input and outputs the optimal action a_i . However, since actor models have a certain failure rate, an Emergency Avoidance Scheme (EAS) is needed at the control level during online execution to complement the actor models to make up for the failure rate.

在执行过程中只使用演员模型。演员 i 仅将其局部观察 o_i 作为输入，并输出最优动作 a_i 。然而，由于演员模型具有一定的失败率，因此在在线执行的控制级别上需要一个紧急回避方案 (EAS) 来补充演员模型，以弥补失败率。

Let $d_i^{obs}\left(a_i\right)$ and $d_i^{v2v}\left(a_i\right)$ be the distances from UAV i to any obstacle and any neighbor at the beginning of the next control loop respectively if the UAV takes action a_i in the current control loop. Since each UAV knows its current position, action and neighboring environments, $d_i^{obs}\left(a_i\right)$ and $d_i^{v2v}\left(a_i\right)$ are easy to predict. A policy action is infeasible if

设 $d_i^{obs}\left(a_i\right)$ 和 $d_i^{v2v}\left(a_i\right)$ 分别为无人机 i 在当前控制循环开始时采取动作 a_i 后到任何障碍物和任何邻居的距离。由于每个无人机都知道其当前位置、动作和邻近环境，$d_i^{obs}\left(a_i\right)$ 和 $d_i^{v2v}\left(a_i\right)$ 很容易预测。如果一个策略动作不可行，则

$$
d_i^{obs}\left(a_i\right) \leq d^{obs} \text{ or } d_i^{v2v}\left(a_i\right) \leq d^{v2v}, \tag{7}
$$

where d^{obs} and d^{v2v} are the safeguard distances to obstacles and neighbors, respectively.

其中 d^{obs} 和 d^{v2v} 分别是到障碍物和邻居的安全距离。

If the action selected the actor model is infeasible, a set of candidate actions are generated without duplicates with a Gaussian noise centered on the policy action:

如果选定的动作使执行器模型不可行，则在没有重复的情况下，以策略动作为中心的高斯噪声生成一组候选动作:

$$
\widetilde{\mathcal{A}}_i \sim \mathcal{N}\left(a_i, \sigma_{\text{ exe }}\right) \tag{8}
$$

The infeasible actions are first excluded from $\widetilde{\mathcal{A}}_i$ based on Eq. 7). The final avoiding action $\widetilde{a}_i$ is then selected as the nearest candidate to a_i in $\widetilde{\mathcal{A}}_i$.

首先根据公式 7) 从 $\widetilde{\mathcal{A}}_i$ 中排除不可行动作。然后，最终避障动作 $\widetilde{a}_i$ 被选为 $\widetilde{\mathcal{A}}_i$ 中距离 a_i 最近的候选动作。

VI. PERFORMANCE EVALUATION

VI. 性能评估

We first present the simulation environment designed to train MACA and then discuss the testing results in comparison with other algorithms. The code for MACA and MACAEnv are available at https://github.com/ OtagoCSSystemsGroup/MACA

我们首先介绍用于训练 MACA 的仿真环境，然后讨论与其他算法相比的测试结果。MACA 和 MACAEnv 的代码可在 https://github.com/OtagoCSSystemsGroup/MACA 上获得。

A. Simulation environment - MACAEnv

A. 仿真环境 - MACAEnv

Existing MARL environments such as StarCraft II [21], Multiple Particle Environment [22] or Multi-Robot Warehouse [23] cannot closely mimic the multi-UAV applications. A new multi-agent environment, MACAEnv, is developed to evaluate the performance of our algorithm and other algorithms. MACAEnv is an episodic environment. An episode starts when a swarm of UAVs and obstacles are spawned and ends when either collision occurs or the obstacles fly through the screen without collisions. For each step in an episode, all UAVs take actions synchronously.

现有的多智能体强化学习环境，如 StarCraft II [21]、多粒子环境 [22] 或多机器人仓库 [23] 无法紧密模拟多无人机应用。为了评估我们的算法和其他算法的性能，开发了一个新的多智能体环境，MACAEnv。MACAEnv 是一个分集环境。当一个无人机群和障碍物生成时，一个剧集开始，当发生碰撞或障碍物无碰撞飞过屏幕时，一个剧集结束。在每一剧集中的每一步，所有无人机同步采取行动。

The environment is shown in Fig. 2. The screen is a 300 m $\times$ 300 m square. At any time, a UAV can only observe a circle window with radius $d = 50$ m around itself. So, our agents are quite short sighted to make sure the environment is partially observed. When an episode is initialized, a swarm with N UAVs is spawned around the middle left of the screen. The positions of UAVs $\mathbf{x}_s$ are equally spaced on a circle with radius of r . The setting of r should match with N , such that the distances between any two UAVs are larger than d^{v2v} initially. For example, we set $r = 30$ m for $N = 3$ and $r = 50$ m for $N = 4$. The initial paths of UAVs are horizontal lines starting from their initial positions pointing to the right most of the screen. The initial speed for each UAV is set to 5 m/s . At the same time, M obstacles are spawned around the middle right, top or bottom side of the screen and are set to fly along straight lines pointing toward the swarm with a velocity of 5 m/s . The initial distance between any UAV and any obstacle is larger than $d = 50$ m so that all obstacles are out of the observation window of each UAV initially. In each training step, a Gaussian noise $\mathcal{N}(0, 0.1)$ is added to the selected action with probability ϵ , where ϵ is annealed from 1.0 to 0.1 in $50k$ training steps to encourage UAVs to explore better actions and avoid local optima.

环境如图 2 所示。屏幕是一个 300 m $\times$ 300 m 正方形。在任何时刻，无人机只能观察到以自身为中心、半径为 $d = 50$ m 的圆形窗口。因此，我们的代理视野非常有限，以确保环境只是部分可观察。当一幕初始化时，一个包含 N 无人机的群在屏幕左中附近生成。无人机 $\mathbf{x}_s$ 的位置均匀地分布在一个半径为 r 的圆上。r 的设置应与 N 匹配，以便任何两个无人机之间的初始距离都大于 d^{v2v} 。例如，我们为 $r = 30$ m 设置 $N = 3$ ，为 $r = 50$ m 设置 $N = 4$ 。无人机的初始路径是从它们的初始位置开始指向屏幕最右端的水平线。每个无人机的初始速度设置为 5 m/s 。同时，M 障碍物在屏幕的右中、顶部或底部附近生成，并设置为以 5 m/s 的速度沿着指向无人机群的直线飞行。任何无人机与任何障碍物之间的初始距离都大于 $d = 50$ m ，以便所有障碍物最初都在每个无人机的观察窗口之外。在每个训练步骤中，以 ϵ 的概率将高斯噪声 $\mathcal{N}(0, 0.1)$ 添加到所选动作中，其中 ϵ 从 1.0 逐渐降温到 0.1，跨越 $50k$ 训练步骤，以鼓励无人机探索更好的动作并避免局部最优。

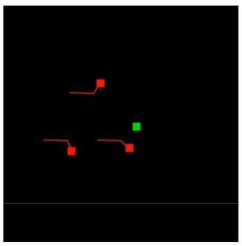


Fig. 2: Snapshots of MACAEnv where red and green squares represent UAVs and obstacles, respectively. The red curves represent trajectories of UAVs.

图 2:MACAEnv 的快照，其中红色和绿色方块分别代表无人机和障碍物。红色曲线代表无人机的轨迹。

B. Learning Performance

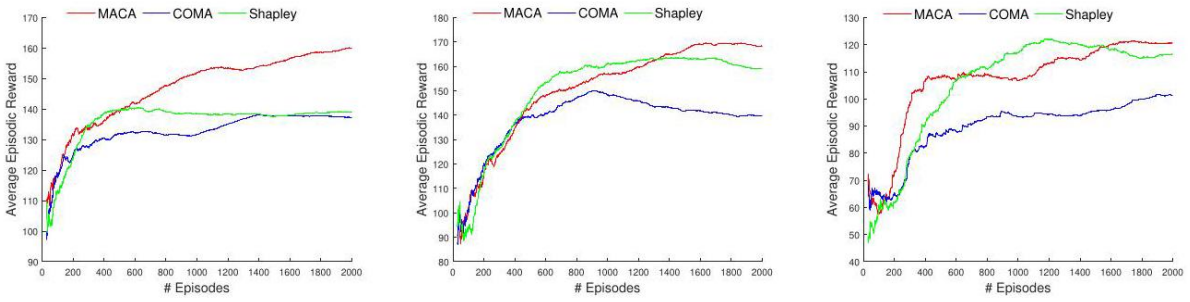
B. 学习性能

We compare MACA with two state-of-the-art algorithms: COMA [7] and Shapley [8].

我们将 MACA 与两种最先进的算法进行了比较:COMA [7] 和 Shapley [8]。

- COMA computes a counterfactual baseline by marginalizing out an agent's action and averaging over all possible actions in the discrete action space. To compare with COMA in our continuous actions space, we add a Gaussian noise to agents' actions. So, a_i' in the advantage function in COMA becomes $a_i' \sim \mathcal{N}(a_i, \sigma)$, where $\sigma = 0.1$.

- COMA 通过边际化出代理的行为并离散动作空间内所有可能的动作求平均来计算一个反事实基线。为了在连续动作空间中与 COMA 进行比较，我们在代理的动作上添加了高斯噪声。因此，COMA 中优势函数的 a_i' 变为 $a_i' \sim \mathcal{N}(a_i, \sigma)$ ，其中 $\sigma = 0.1$ 。



(a) Results of scenario *2U-1O*. (b) Results of scenario *3U-1O*. (c) Results of scenario *4U-2O*.

Fig. 3: Learning performance of all three algorithms and parameter analysis of MACA.

图 3: 所有三种算法的学习性能以及 MACA 的参数分析。

- Shapley computes a baseline for each agent by averaging over all possible collations that include this agent. Its complexity grows with the increase of swarm size.

- Shapley 通过平均包含该代理的所有可能的协作来为每个代理计算一个基线。其复杂度随着群体规模的增加而增加。

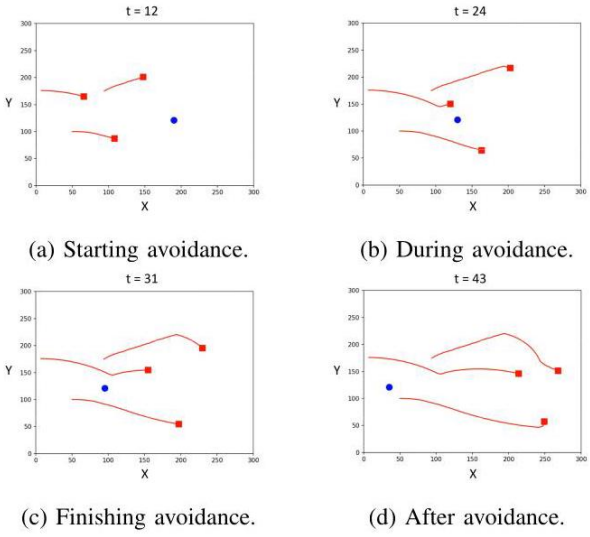
We perform tests in three scenarios: 2-UAV-1-Obstacle (2U10), 3-UAV-1-Obstacle (3U10) and 4-UAV-2-Obstacle (4U2O). We choose small-size swarms in our experiments because the training takes very long time. For example, it takes more than 10 hours to train in any of our scenarios. For larger-size swarms it would take weeks to train. We will provide results for larger swarms in the final version, though the results on small sizes have already validated our theoretical analysis.

我们在三种场景下进行测试:2-UAV-1-Obstacle (2U10)、3-UAV-1-Obstacle (3U10) 和 4-UAV-2-Obstacle (4U2O)。我们在实验中选择小规模群体，因为训练需要非常长的时间。例如，在我们的任何一种场景中，训练都需要超过 10 小时。对于更大规模的群体，训练可能需要数周时间。尽管小规模的实验结果已经验证了我们的理论分析，我们将在最终版本中提供更大规模群体的结果。

In the following experiments, we set $C = 0$ and $D^{\max} = 150$ in the reward function for MACA, $\sigma_{\text{exe}} = 1.0$ for EAS. The collision penalty C can be set to 0 in MACAEnv, because the episode ends once collision occurs. The action causing collision is penalized by ending of game. The test results are shown in Fig. 3. In all scenarios, our scheme achieves the highest reward, COMA achieves the lowest and Shapley ranks in the middle. This is because our scheme doesn't need to select any default action when calculating the counterfactual baseline. The contribution of each agent is purely calculated by the

collection of observations and actions from all agents except that agent, which just depends on the current state of the environment. However, the counterfactual baseline in COMA is calculated by averaging over all possible actions in the agent's action space. This counterfactual baseline is representative enough in discrete action spaces where one can loop over all possible actions to calculate the corresponding Q values. But it is not representative enough in continuous action space where the action space is sampled based on a distribution model. In Shapley, the counterfactual baseline is calculated by replacing an agent's action with a default action and then averaging over all possible collations that include the agent per se. When the number of agents is small like three or four, the counterfactual baseline in Shapley is not representative enough and the performance is poor. Fig. 3 also shows that the highest rewards all algorithms can achieve in scenario $4U2O$ are much lower than those in scenario $2U10$ and $3U10$. That is because the environment is much more complex in scenario $4U2O$ and its joint action space is also much larger, making it difficult to find the optimal joint actions to achieve high global reward. Overall, our algorithm achieves an average reward that is better (16.61%, 20.41% and 19.22% higher) than the other two algorithms in scenario $2U10, 3U10$ and $4U2O$, respectively.

在以下实验中，我们为 MACA 的奖励函数设置了 $C = 0$ 和 $D^{\max} =$ 为 150，为 EAS 设置了 $\sigma_{\text{exe}} = 1.0$ 。在 MACAEnv 中，碰撞惩罚 C 可以设置为 0，因为一旦发生碰撞，剧集就会结束。导致碰撞的动作会被游戏结束所惩罚。测试结果如图 3 所示。在所有场景中，我们的方案获得了最高的奖励，COMA 获得了最低的奖励，而 Shapley 排名居中。这是因为我们的方案在计算反事实基线时不需要选择任何默认动作。每个代理的贡献纯粹是通过收集除该代理外的所有代理的观察和动作来计算的，这仅依赖于环境的当前状态。然而，COMA 中的反事实基线是通过计算代理动作空间中所有可能动作的平均值来得到的。这种反事实基线在离散动作空间中足够代表性，因为可以遍历所有可能的动作来计算相应的 Q 值。但在连续动作空间中，动作空间是基于分布模型进行采样的，这种反事实基线代表性不足。在 Shapley 中，反事实基线是通过将代理的动作替换为默认动作，然后对所有包含该代理的可能组合进行平均来计算的。当代理数量较少，如三个或四个时，Shapley 中的反事实基线代表性不足，性能较差。图 3 还显示，所有算法在场景 $4U2O$ 中能获得的最大奖励远低于场景 $2U10$ 和 $3U10$ 中的奖励。这是因为场景 $4U2O$ 中的环境更为复杂，其联合动作空间也更大，使得找到实现高全局奖励的最优联合动作变得困难。总体而言，我们的算法在场景 $2U10, 3U10$ 和 $4U2O$ 中分别获得了比其他两种算法平均奖励高出 16.61%、20.41% 和 19.22% 的成绩。



(a) Starting avoidance. (b) During avoidance.

(c) Finishing avoidance. (d) After avoidance.

Fig. 4: Demonstrations of trajectories generated by MACA.
图 4:MACA 生成的轨迹演示。

C. Trajectory Demonstration

C. 轨迹演示

Fig. 4 shows the trajectories generated by MACA with three UAVs and one obstacle. The UAVs and obstacles all have a velocity of 5 m/s , flying towards each other. The red squares and blue dots represent

UAVs and obstacles, respectively. The red curves represent trajectories of UAVs. A short video is also attached with in the submission of this paper to demonstrate the movement and trajectories of UAVs.

图 4 显示了 MACA 使用三架无人机和一个障碍物生成的轨迹。无人机和障碍物都具有 5 m/s 的速度，相互飞向对方。红色正方形和蓝色点分别代表无人机和障碍物。红色曲线代表无人机的轨迹。本文提交中还附有一个短视频，用于演示无人机的运动和轨迹。

D. Safety, Energy and Response time

D. 安全性、能耗和响应时间

In this experiment, we compare the safety, energy consumption and response time of MACA, COMA and Shapley against a state-of-the-art conventional algorithm E^2 Coop [3]. Safety is evaluated by the minimum distance among swarm members and the minimum distance between a UAV and an obstacle during avoidance. Energy consumption is calculated using the energy model proposed in E^2 Coop. For a fair comparison, the power consumption of E^2 Coop also includes the energy spent on communication since it highly relies on the vehicle-to-vehicle communication. We assume each UAV uses a WIFI transceiver and the communication power is set to $2W$ according to the off-the-shelf WIFI radio chips [24]. The weight of a UAV is set to 1Kg.d^{obs} and d^{v2v} are set to 20 m and 5 m , respectively. We compare our algorithm against E^2 Coop under the setting $\theta = 45^\circ$, as it is empirically recommended by our simulations. We also did ablation tests to validate the importance of EAS in decentralized execution. The results are shown in Fig. 5 and all schemes are performed 100 episodes.

在本次实验中，我们比较了 MACA、COMA 和 Shapley 与最先进的传统算法 E^2 Coop [3] 的安全性、能耗和响应时间。安全性通过群成员之间的最小距离以及无人机在避障过程中的最小距离来评估。能耗使用 E^2 Coop 中提出的能量模型进行计算。为了公平比较，E^2 Coop 的功耗还包括了在车辆间通信上消耗的能量，因为它高度依赖于车与车之间的通信。我们假设每架无人机使用 WIFI 收发器，通信功率根据现成的 WIFI 射频芯片 [24] 设置为 $2W$ 。无人机的重量设置为 1Kg.d^{obs} ，d^{v2v} 分别设置为 20 m 和 5 m 。我们在 $\theta = 45^\circ$ 的设置下与 E^2 Coop 进行比较，因为我们的模拟实验推荐使用这种设置。我们还进行了消融测试，以验证 EAS 在去中心化执行中的重要性。结果如图 5 所示，所有方案都执行了 100 个回合。

	2U10				3U10				4U20			
	Safety (m) d^{obs} , d^{v2v}	Failure rate	Energy ($\times 10^3 J$)	Respond time (s)	Safety (m) d^{obs}, d^{v2v}	Failure rate	Energy ($\times 10^3 J$)	Respond time (s)	Safety (m) d^{obs}, d^{v2v}	Failure rate	Energy ($\times 10^3 J$)	Respond time (s)
E^2 Coop	34.52 ± 0.18 , 39.93 ± 0.18	0%	1.2974	0.3673	24.06 ± 0.12 , 29.20 ± 0.02	0%	1.2950	0.4666	21.07 ± 4.01 , 24.18 ± 1.82	21%	1.4245	0.6272
Shapley	47.59 ± 20.72 94.99 ± 13.16	3%	2.6661	0.0065	35.33 ± 10.48 55.42 ± 11.84	7%	3.0013	0.0070	59.82 ± 44.27 51.94 ± 9.71	7%	2.2731	0.0091
COMA	59.19 ± 24.16 99.83 ± 1.03	0	2.5680	0.0068	59.19 ± 24.16 99.83 ± 1.03	0	2.5680	0.0068	25.40 ± 2.54, 44.89 ± 3.63	14%	3.9619	0.0063
MACA(no E)	$54.67 + 30.86$ $97.56 + 8.02$	0	1.5658	0.0071	$44.14 + 16.32$. $63.16 + 14.37$	2%	3.0953	0.0090	$69.37 + 27.71$. 42.51 ± 1.91	12%	2.9142	0.0093
MACA	$70.25 + 35.06$ $97.67 + 6.35$	0	1.6050	0.0034	$44.16 + 21.30$ $81.58 + 9.74$	0	2.0437	0.0032	$38.09 + 20.21$ $43.46 + 2.51$	2%	2.6912	0.0071

	2U10				3U10				4U20			
	安全性 (米) d^{obs} , d^{v2v}	失败率	能量 ($\times 10^3 J$)	响应时间 (秒)	安全性 (米) d^{obs}, d^{v2v}	失败率	能量 ($\times 10^3 J$)	响应时间 (秒)	安全性 (米) d^{obs}, d^{v2v}	失败率	能量 ($\times 10^3 J$)	响应时间 (秒)
E^2 Coop	34.52 ± 0.18 , 39.93 ± 0.18	0%	1.2974	0.3673	24.06 ± 0.12 , 29.20 ± 0.02	0%	1.2950	0.4666	21.07 ± 4.01 , 24.18 ± 1.82	21%	1.4245	0.6272
Shapley	47.59 ± 20.72 94.99 ± 13.16	3%	2.6661	0.0065	35.33 ± 10.48 55.42 ± 11.84	7%	3.0013	0.0070	59.82 ± 44.27 51.94 ± 9.71	7%	2.2731	0.0091
COMA	59.19 ± 24.16 99.83 ± 1.03	0	2.5680	0.0068	59.19 ± 24.16 99.83 ± 1.03	0	2.5680	0.0068	25.40 ± 2.54, 44.89 ± 3.63	14%	3.9619	0.0063
MACA(无 E)	$54.67 + 30.86$ $97.56 + 8.02$	0	1.5658	0.0071	$44.14 + 16.32$. $63.16 + 14.37$	2%	3.0953	0.0090	$69.37 + 27.71$. 42.51 ± 1.91	12%	2.9142	0.0093
MACA	$70.25 + 35.06$ $97.67 + 6.35$	0	1.6050	0.0034	$44.16 + 21.30$ $81.58 + 9.74$	0	2.0437	0.0032	$38.09 + 20.21$ $43.46 + 2.51$	2%	2.6912	0.0071

Fig. 5: Comparison of MACA with state-of-the-art MARL algorithms and conventional algorithms.

图 5:MACA 与最先进的 MARL 算法和传统算法的比较。

It can be seen from Fig. 5 that all MARL algorithms greatly reduce the respond time and achieve lower failure rate compared to E^2 Coop. The energy consumption of the three MARL algorithms are slightly higher than E^2 Coop. This is because the cost function in E^2 Coop penalizes UAVs for deviating from the trajectories that are centrally planned, whereas in MARL algorithms there is no central planning and all trajectories are searched individually with safety being the highest priority. Therefore, UAVs in MARL algorithms try to maintain a relative large distance to neighbors and obstacles. While each UAV in E^2 Coop tends to keep a distance close to the safeguard distances while ensuring safety. This also explains why the distances between UAVs and obstacles are generally larger in MARL algorithms than in E^2 Coop. Comparing with MACA(noEAS) and the two other MARL algorithms, MACA(noEAS) generally achieves a lower failure rate and consumes less energy. This shows that the counterfactual baseline in our algorithm better represents the importance of each agent, and therefore the agents in our algorithm learn more robust policies.

从图 5 可以看出，所有多智能体强化学习 (MARL) 算法大大减少了响应时间，并且相比 E^2 Coop，实现了更低的失败率。这三个 MARL 算法的能量消耗略高于 E^2 Coop。这是因为 E^2 Coop 中的成本函数对无人机偏离中央规划的轨迹进行了惩罚，而在 MARL 算法中不存在中央规划，所有轨迹都是单独搜索的，安全是最高优先级。因此，在 MARL 算法中的无人机试图与邻居和障碍物保持相对较大的距离。而 E^2 Coop 中的每架无人机倾向于在确保安全的同时保持接近安全距离。这也解释了为什么在 MARL 算法中无人机与障碍物之间的距离通常比 E^2 Coop 中的要大。与 MACA(noEAS) 和其他两种 MARL 算

法相比，MACA(noEAS) 通常实现了更低的失败率和更低的能耗。这表明我们算法中的反事实基线更好地代表了每个智能体的重要性，因此我们算法中的智能体学习了更加鲁棒的政策。

The biggest advantage of MACA is its short response time for each step. The solutions are found almost instantly by a simple forward pass of the actor models. While in E^2 Coop, the solutions are found by repeatedly performing the whole algorithm at each step, considering neighboring UAVs and obstacles. So the respond time of E^2 Coop is much larger than that of MACA and grows with the increase on the number of UAVs. MACA(noEAS) also has a failure rate of 2% and 12% in scenario $3U10$ and $4U2O$, respectively. But with EAS, the failure rate is reduced to 0 and 2% in scenario $3U10$ and $4U2O$, respectively. Overall, our algorithm reduces respond time by over 99% compared to E^2 Coop in all scenarios. We reduce the failure rate by 90% in scenario $4U2O$, respectively, even though the energy consumption of MACA increases a little.

MACA 最大的优势在于每一步的响应时间很短。解决方案几乎可以立即通过演员模型的简单前向传递找到。而在 E^2 Coop 中，解决方案是通过在每一步重复执行整个算法来找到的，同时考虑相邻的无人机和障碍物。因此，E^2 Coop 的响应时间比 MACA 长得多，并且随着无人机数量的增加而增加。MACA(noEAS) 在场景 $3U10$ 和 $4U2O$ 中的失败率分别为 2% 和 12% 。但是使用 EAS 后，失败率分别降低到 0 和 2% 。总体而言，我们的算法在所有场景中都将响应时间缩短了超过 99% ，与 E^2 Coop 相比。我们在场景 $4U2O$ 中将失败率降低了 90% ，尽管 MACA 的能量消耗略有增加。

VII. CONCLUSION

VII. 结论

In this paper, a new multi-actor centralized critic learning scheme named MACA is proposed for collision avoidance of UAV swarms. A new counterfactual advantage function with low complexity is proposed and this advantage function can also apply to continuous action spaces. An emergency avoidance scheme is proposed to accommodate the failure rate of the actor models in execution phase. Comprehensive experiments are conducted in our environment MACAEnv to show the advantage of our algorithm. Future work will focus on further reduction of energy consumption in MACA.

在本文中，我们提出了一种名为 MACA 的新多演员集中式评判学习方案，用于无人机群体的避障。提出了一种新的低复杂度反事实优势函数，该优势函数也可应用于连续动作空间。我们还提出了一种紧急避障方案，以应对执行阶段演员模型的失败率。在我们的环境 MACAEnv 中进行了全面实验，以展示我们算法的优势。未来的工作将专注于进一步降低 MACA 的能量消耗。

REFERENCES

参考文献

[1] K. Jose and D. K. Pratihar, "Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods," Robotics and Autonomous Systems, vol. 80, pp. 34-42, 2016. [Online]. Available: https: //www.sciencedirect.com/science/article/pii/S0921889016000282

[2] C. Liang, X. Zhang, Y. Watanabe, and Y. Deng, "Autonomous collision avoidance of unmanned surface vehicles based on improved a star and minimum course alteration algorithms," Applied Ocean Research, vol. 113, p. 102755, 2021.

[3] S. Huang, H. Zhang, and Z. Huang, "E2coop: Energy efficient and cooperative obstacle detection and avoidance for uav swarms," in Proceedings of the International Conference on Automated Planning and Scheduling, vol. 31, 2021, pp. 634-642.

[4] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls et al., "Value-decomposition networks for cooperative multi-agent learning," arXiv preprint arXiv:1706.05296, 2017.

[5] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," 2018.

[6] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," 2019.

[7] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018.

[8] J. Li, K. Kuang, B. Wang, F. Liu, L. Chen, F. Wu, and J. Xiao, "Shapley counterfactual credits for multi-agent reinforcement learning," Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Aug 2021. [Online]. Available: http://dx.doi.org/10.1145/3447548.3467420

[9] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," IEEE Transactions on Robotics, vol. 27, no. 4, pp. 696-706, 2011.

[10] S. Huang and K. Low, "A path planning algorithm for smooth trajectories of unmanned aerial vehicles via potential fields," in 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). IEEE, 2018, pp. 1677-1684.

[11] B. Song, Z. Wang, L. Zou, L. Xu, and F. E. Alsaadi, "A new approach to smooth global path planning of mobile robots with kinematic constraints," International Journal of Machine Learning and Cybernetics, vol. 10, no. 1, pp. 107-119, 2019.

[12] Y. Kuriki and T. Namerikawa, "Formation control with collision avoidance for a multi-uav system using decentralized mpc and consensus-based control," SICE Journal of Control, Measurement, and System Integration, vol. 8, no. 4, pp. 285-294, 2015.

[13] D. Wang, T. Fan, T. Han, and J. Pan, "A two-stage reinforcement learning approach for multi-uav collision avoidance under imperfect sensing," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 3098-3105, 2020.

[14] Y.-H. Hsu and R.-H. Gau, "Reinforcement learning-based collision avoidance and optimal trajectory planning in uav communication networks," IEEE Transactions on Mobile Computing, 2020.

[15] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized noncommunicating multiagent collision avoidance with deep reinforcement learning," in 2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017, pp. 285-292.

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

[17] D. H. Wolpert and K. Tumer, "Optimal payoff functions for members of collectives," in Modeling complexity in economic and social systems. World Scientific, 2002, pp. 355-369.

[18] SICK, "Ld-mrs420201," https://www.sick.com/de/en/ detection-and-ranging-solutions/3d-lidar-sensors/ld-mrs/c/g91913 2021, (Accessed on 12/16/2021).

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," nature, vol. 518, no. 7540, pp. 529-533, 2015.

[20] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in Advances in neural information processing systems, 2000, pp. 1057-1063.

[21] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser et al., "Starcraft ii: A new challenge for reinforcement learning," arXiv preprint arXiv:1708.04782, 2017.

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

[23] G. Papoudakis, F. Christianos, L. Schäfer, and S. V. Albrecht, "Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks," in Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS), 2021. [Online]. Available: http://arxiv.org/abs/2006.07869

[24] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11n power consumption," in Proceedings of the 2010 international conference on Power aware computing and systems. USENIX Association, 2010, p. 1.