A Multi-Agent Reinforcement Learning Approach to Traffic Control at Future Urban Air Mobility Intersections

未来城市空中出行交叉点的多代理强化学习交通控制方法

Sabrullah Deniz1
Sabrullah Deniz1
University of Tennessee, Knoxville, Tennessee 37996, USA
田纳西大学，诺克斯维尔，田纳西州 37996，美国
Zhenbo Wang2
Zhenbo Wang2
University of Tennessee, Knoxville, Tennessee 37996, USA
田纳西大学，诺克斯维尔，田纳西州 37996，美国

Today, air traffic controllers communicate with pilots via radio to direct the aircraft. The increasing demand in drone delivery and air mobility will increase air density, which requires highly automated air traffic control systems. To meet the growing demand in air transportation, a high standard autonomous support system is needed. In particular, we need an autonomous air traffic controller to keep the airspace safe and efficient. In this study, we propose a novel Multi-Agent Reinforcement Learning (MARL) approach to handle high-density UAM operations by providing effective guidance to electric vertical takeoff and landing (eVTOL) vehicles to avoid traffic congestion and reduce travel time. The goal of our MARL approach is to reduce the time at the environed urban air intersections by providing the speed advisories to each approaching vehicle for safe separation at the intersection. The proposed model is trained and evaluated in BlueSky, an open-source air traffic control simulation environment. The results of our simulations with real-world data from thousands of aircraft show that using MARL for the separation problem at the intersection is very promising for solving the problem of en-route air traffic control.

如今，空中交通管制员通过无线电与飞行员通信来指挥飞机。无人机送货和空中出行的需求日益增长，将增加空气密度，这需要高度自动化的空中交通控制系统。为了满足空中交通日益增长的需求，需要一个高标准的自主支持系统。特别是，我们需要一个自主的空中交通管制员来保持空域的安全和高效。在本研究中，我们提出了一种新颖的多代理强化学习 (MARL) 方法来处理高密度的城市空中出行 (UAM) 业务，通过为电动垂直起降 (eVTOL) 车辆提供有效指导，以避免交通拥堵和减少旅行时间。我们的 MARL 方法的目标是通过为每个接近交叉点的车辆提供速度建议，以实现交叉点的安全分离，从而减少在包围的城市空中交叉点的时间。所提出的模型在 BlueSky 中进行了训练和评估，BlueSky 是一个开源的空中交通控制仿真环境。我们使用现实世界中数千架飞机的数据进行的模拟结果表明，使用 MARL 解决交叉点的分离问题对于解决航路空中交通控制问题非常有希望。

I. Introduction

I. 引言

A. General Motivation

A. 一般动机

Unmanned aerial vehicles (UAVs) recently have been used for many different purposes such as package delivery, agriculture spraying, fire detection, emergency response, and weather forecasting. It is expected that millions of UAVs will be in U.S. airspace by 2040, according to the Federal Aviation Administration (FAA) report [1]. In order to maintain the airspace safety, the traffic management of a large number of UAVs is a crucial problem that needs to be solved in order to reduce the airspace traffic congestion. Current air traffic controller (ATC), which depends on human controllers to manage air traffic, is more expensive and unsafe due to needs of human multitasking for this massive traffic management. In 2018, the FAA NextGen Office released an initial Concept of Operations (ConOps) for unmanned air traffic system (UAS) traffic management (UTM) that is defined for support in lower (UAS) operations. The FAA and NASA NextGen have updated that ConOps to continue maturing the more complex UTM operations as v2.0 ConOps [2] in order to increase the airspace capacity. Despite substantial development, there are several research issues that remain unresolved [3][4]. In this research, our aim is to address one of

the research challenges, which is how to manage the traffic at the intersections for multiple UAVs in a large-scale environment inside the envisioned Air Corridors [5] or UAM (urban air mobility) Corridors is proposed by FAA and NASA as shown in Fig. 1 below. UAM Corridors are safe and efficient during the high-density air traffic missions. The traffic management inside those corridors is our main research focus in this paper.

无人驾驶飞行器 (UAVs) 最近已被用于多种不同目的，例如包裹递送、农业喷洒、火灾探测、紧急响应和天气预报。根据美国联邦航空管理局 (FAA) 的报告，预计到 2040 年，美国空域将有一百万架 UAVs。为了维护空域安全，大量 UAV 的交通管理是一个关键问题，需要解决以减少空域交通拥堵。当前的空中交通管制 (ATC)，依赖于人工管制员来管理空中交通，由于需要人工进行大规模交通管理，成本更高且不安全。2018 年，FAA NextGen 办公室发布了一份针对无人机交通系统 (UAS) 交通管理 (UTM) 的初步操作概念 (ConOps)，该操作概念定义为支持低空 (UAS) 操作。FAA 和 NASA NextGen 更新了该 ConOps，以继续完善更复杂的 UTM 操作，作为 v2.0 ConOps [2]，以增加空域容量。尽管取得了重大发展，但仍有一些研究问题尚未解决 [3][4]。在这项研究中，我们的目标是解决其中一个研究挑战，即如何在如图 1 所示的 FAA 和 NASA 提出的设想空中走廊 [Air Corridors] 或城市空中移动 (UAM) 走廊内，大规模环境中为多个 UAV 管理交叉路口的交通。

Fig. 1 UAM Corridors proposed by NASA [6].

图 1 美国宇航局 [6] 提出的 UAM 走廊。

Since urban air mobility (UAM) becomes more promising for future transportation, many companies are forming UAM concepts including NASA, Uber, Airbus, Volocopter, Bell, and Embraer [7][8][9][10]. This concept is known as electric vertical takeoff and landing (eVTOL) aircraft which uses three-dimensional airspace for personal commute or on-demand air taxi. The first published Concept of Operations V1.0 by FAA and NASA NextGen for urban air mobility stated that the UAM will be traveling in an air corridor defined and declared by the FAA but mostly modified by the industry and other stakeholders. To travel in the corridor, eVTOL will have to meet its performance requirements and follow the restrictions, which can be different based on surrounding airspace and from Corridor to Corridor [5]. Along the route, the corridors are going to either merge or have intersections. ATC will assess the availability of Corridors based on surrounding circumstances and activities, but tactical separation facilities will not be provided within them, and safe operations will be ensured by pilots in command [5]. For fully autonomous piloting and flight, this decision is assumed to be made by a well-trained reinforcement learning agent in this paper. Reinforcement learning is a model-free self-learning algorithm that offers new solutions to air traffic management and intersection control. The aim of reinforcement learning is to allow an agent to learn the best policy by interacting with its surroundings and maximize the cumulative reward. The reward

[1] Ph.D. student, Department of Mechanical, Aerospace, and Biomedical Engineering.
[1] 博士研究生，机械、航空航天和生物医学工程系。
[2] Assistant Professor, Department of Mechanical, Aerospace, and Biomedical Engineering, AIAA Member.
[2] 助理教授，机械、航空航天和生物医学工程系，AIAA 会员。

is the feedback from the environment and rewards the agent for good behavior while punishing for bad behavior. We can get dynamic real-time conflict resolution advisories to the aircraft by formulating the tasks of human air traffic controllers as a reinforcement learning problem with a little computation time.

由于城市空中出行 (UAM) 在未来交通中显示出越来越大的潜力，许多公司包括美国宇航局、Uber、Airbus、Volocopter、Bell 和 Embraer [7][8][9][10] 正在形成 UAM 概念。这个概念被称为电动垂直起降 (eVTOL) 飞机，它使用三维空域进行个人通勤或按需空中出租车服务。美国联邦航空管理局 (FAA) 和美国宇航局 NextGen 发布的第一个城市空中出行概念操作 V1.0 指出，UAM 将在 FAA 定义和宣布的空中走廊中行驶，但主要由行业和其他利益相关者进行修改。要在走廊中行驶，eVTOL 将必须满足其性能要求并遵守限制，这些限制可能会根据周围的空域和走廊之间的不同而有所不同 [5]。在航线沿途，走廊将会合并或相交。空中交通管制 (ATC) 将根据周围环境和活动评估走廊的可用性，但战术分离设施不会在它们内部提供，安全运行将由机长负责 [5]。对于完全自主驾驶和飞行，本文假设这一决定将由训练有素的强化学习代理做出。强化学习是一种无需模型的自我学习算法，为空中交通管理和交叉口控制提供了新的解决方案。强化学习的目标是让代理通过与周围环境的交互学习最佳策略，并最大化累积奖励。奖励是来自环境的反馈，对代理的良好行为给予奖励，对不良行为进行惩罚。通过将人为空中交通管制员的任务公式化为一个强化学习问题，我们可以在很小的计算时间内获得动态实时冲突解决建议，以供飞机使用。

Reinforcement learning has recently received attention due to the performance of the deep reinforcement learning agent AlphaGo [11] and OpenAI [12]. AlphaGo is the first computer program to beat a skilled human Go player, as well as the first to defeat a Go world champion. This significant advancement in AI demonstrated the theoretical basis and computational capabilities of intelligent agents and AI technologies to potentially complement and promote human tasks. BlueSky air traffic control simulator [13], a fast-time simulator that utilizes such techniques by allowing the agent to interact with the environment, is used to demonstrate the model performance.

由于深度强化学习代理 AlphaGo [11] 和 OpenAI [12] 的性能表现，强化学习最近受到了关注。AlphaGo 是第一个击败熟练人类围棋选手以及围棋世界冠军的计算机程序。这一人工智能领域的重大进步展示了智能代理和人工智能技术的理论基础和计算能力，这些技术有潜力补充和促进人类任务。BlueSky 空中交通管制模拟器 [13]，一个快速时间模拟器，通过允许代理与环境互动来利用这些技术，用于展示模型性能。

In this work, a deep multi-agent reinforcement learning (MARL) framework is used to enable autonomous air traffic separation in en-route airspace to reduce conflicts at intersections and during the flight by using an approach of decentralized execution and centralized learning. The proposed framework can handle a variable of aircraft at the intersections by providing speed advisories for each aircraft along the route. Long Short-Term Memory (LSTM) networks [14] are used to store information about the environment into a fixed length vector which gives all information about the environment. This proposed work has a high potential to solve an autonomous air traffic intersection management problem with air corridors.

在这项工作中，使用深度多代理强化学习 (MARL) 框架，实现在航路空域中的自主空中交通分离，以减少交叉点和飞行过程中的冲突，采用了一种去中心化执行和集中式学习的方法。该框架能够通过为沿途每架飞机提供速度建议来处理交叉点的变量飞机。长短期记忆 (LSTM) 网络 [14] 用于将环境信息存储到固定长度向量中，该向量提供了关于环境的所有信息。这项提议的工作有很大的潜力解决带有空中走廊的自主空中交通交叉点管理问题。

B. Related Work

B. 相关工作

Deep reinforcement learning has already been used for traffic light control in ground transportation to reduce long delay of travelers and provide safe autonomous passes at intersections [15][16]. In the ground transportation, the agent is placed at the intersection to control the traffic lights to reduce the waiting time. While in the ground transportation, the agent controls the lights for ground vehicles to stop and go, in our research we provide the speed advisory to each eVTOL vehicle to avoid any conflict at intersections. The major difference in our research is that each eVTOL vehicle represents an agent and is provided with speed advisories to handle potential conflicts at the intersection.

深度强化学习已经用于地面交通中的交通灯控制，以减少旅客的长时延误并提供交叉口的自主安全通行 [15][16]。在地面交通中，代理被放置在交叉口控制交通灯以减少等待时间。而在地面交通中，代理控制灯光让地面车辆停止和行驶，在我们的研究中，我们为每辆 eVTOL 提供速度建议，以避免交叉点的任何冲突。我们研究的主要区别在于，每辆 eVTOL 车辆代表一个代理，并获得了速度建议，以处理交叉

点的潜在冲突。

Autonomous air traffic control has been studied for a long time. Heinz Erzberger and his NASA colleagues developed an auto-resolver that computes air traffic trajectories iteratively and tests candidate trajectories until a suitable trajectory is found that meets all conflict resolution condition [17][18]. With several iteration loops, the algorithm generates the resolution trajectories. This detection system also includes characteristics of the conflict such as aircraft speed, altitudes, flight plans, and coordinates. By that physic-based auto-resolver approach, conflict detection and conflict resolution have a good performance.

自主空中交通控制已经研究了很长时间。Heinz Erzberger 及其 NASA 同事开发了一种自动解析器，该解析器迭代计算空中交通轨迹，并测试候选轨迹，直到找到一个满足所有冲突解决条件的合适轨迹 [17][18]。通过几个迭代循环，算法生成解析轨迹。这个检测系统还包括冲突特征，如飞机速度、高度、飞行计划和坐标。通过基于物理的自动解析器方法，冲突检测和冲突解决表现出良好的性能。

In another research, the researcher used a multiagent approach by using reinforcement learning with a comprehensive reward function [19]. In that approach, the author considered each agent independently, due to the computational limitation, as a fix location in 2D space that uses three separate actions to navigate the airspace: setting separation between aircrafts, setting the aircraft departures and ground delay, and rerouting aircraft. Then agents use reinforcement learning to learn the best optimal actions. The result of their research shows that agents reduce congestion by up to 80% and some limitations with the learning process show that there are more efficient and effective methods than those approaches cited above.

在另一项研究中，研究人员使用了一个多代理方法，通过使用带有综合奖励函数的强化学习 [19]。在该方法中，作者由于计算限制，将每个代理独立考虑，作为一个在二维空间中的固定位置，使用三种独立动作来导航空域: 设置飞机间的间隔、设置飞机起飞和地面延误，以及重新规划飞机航线。然后代理使用强化学习来学习最佳优化动作。他们的研究结果表明，代理可以减少拥堵高达 80% ，并且学习过程中的某些局限性表明，存在比上述方法更有效和高效的方法。

Recent improvement in deep learning and in computer hardware means that we can compute and test a real-time multi agent policy in a more realistic environment. ATC automation using MARL was formulated more recently [20]. In that work, a deep MARL framework is proposed to handle the separation problem for ATC. Their results show that the model has a very promising performance for ATC problem. In that research, the UAM intersection problem was formulated as an MARL problem and solved by a DD-MARL framework, which has a good performance to solve complex sequential decision-making problems under uncertainty.

深度学习和计算机硬件的最近改进意味着我们可以计算并测试更真实环境中的实时多代理策略。使用 MARL 的 ATC 自动化最近被提出 [20]。在那项工作中，提出了一个深度 MARL 框架来处理 ATC 的分隔问题。他们的结果显示，该模型在 ATC 问题上具有非常有前景的性能。在该研究中，UAM 交叉问题被公式化为一个 MARL 问题，并通过 DD-MARL 框架解决，该框架在解决不确定性的复杂序列决策问题方面表现良好。

In our work, we will use the same MARL model for single intersection with more complex environment and different parameters to make a more realistic model. Our model, as we earlier mentioned, will operate in structured airways or air corridors within the UAM concept proposed by NASA and Uber Elevate [21].

在我们的工作中，我们将对单个交叉路口使用相同的 MARL 模型，并采用更复杂的环境和不同的参数来构建更加现实的模型。正如我们之前提到的，我们的模型将在 NASA 和 Uber Elevate 提出的 UAM 概念下的结构化空域或空中走廊中运行 [21]。

The rest of this paper is as follows. In Section II, reinforcement learning, policy-based learning, and MARL will be introduced. System models and problem formulation are presented in Section III. In Section IV, we will detail our Deep MARL framework. Section V presents the performance of our model, and Section VI concludes this paper.

本文的其余部分如下。在第二部分，我们将介绍强化学习、基于策略的学习和 MARL。第三部分将展示系统模型和问题公式化。在第四部分，我们将详细描述我们的深度 MARL 框架。第五部分将展示我们的模型性能，第六部分将总结本文。

II. Preliminaries

II. 预备知识

A. Reinforcement Learning

A. 强化学习

Reinforcement learning is a type of machine learning that is different from supervised learning and unsupervised learning [22]. Reinforcement learning has received increased attention to solve vehicle coordination and management problems for both ground transportation and air mobility. It interacts with unknown environments to learn the policy by getting rewards from actions. The goal is to maximize the numerical rewards in interactive environments where an agent can take different actions that affect the rewards. At each time step t , an agent observes the current state s_t and selects an action a_t , then the state updated and agent receives a reward r_t . The state evolves from $s_t \rightarrow s_{t+1}$ and the next action a_{t+1} depends on the dynamics of the environment, which is often unknown due to unknown transition probabilities from one state to another in model free environments.

强化学习是一种与监督学习和无监督学习不同的机器学习方法 [22]。强化学习在解决地面交通和空中移动的车辆协调和管理问题方面受到了越来越多的关注。它通过与未知环境交互，通过从行动中获得奖励来学习策略。目标是最大化互动环境中的数值奖励，其中智能体可以采取不同的行动来影响奖励。在每一个时间步 t ，智能体观察当前状态 s_t 并选择一个动作 a_t ，然后状态更新，智能体接收到奖励 r_t 。状态从 $s_t \rightarrow s_{t+1}$ 发展而来，下一个动作 a_{t+1} 取决于环境的动态，这在模型自由的环境中通常是由于状态之间的转移概率未知而未知。

We can denote a reinforcement learning model as S, A, R, T with the following meanings:

我们可以用以下含义来表示一个强化学习模型 S, A, R, T :

- S : A set of all possible states in the environment, s is one of the states $(s \in S)$.
- S : 环境中所有可能状态的集合，s 是其中的一个状态 $(s \in S)$ 。
- A : A set of all possible actions that an agent can choose, a is an action $(a \in A)$.
- A : 智能体可以选择的所有可能动作的集合，a 是一个动作 $(a \in A)$ 。
- R : The reward function decides how much reward an agent should take from one state to another $R\left(s_t, a_t, s_{t+1}\right)$.
- R : 奖励函数决定了智能体从一个状态转移到另一个状态时应获得多少奖励 $R\left(s_t, a_t, s_{t+1}\right)$ 。
- $\gamma \in [0, 1]$ − A discount factor decides based on the performance on immediate and future rewards. For convergence of cumulative reward, the discount factor should be chosen less than 1 . In the above definitions, t represents the current time, while T is a total time and π represents the policy. The goal is to find an optimal policy π^* for any initial state to maximize the overall cumulative rewards for the future steps as follows:
- $\gamma \in [0, 1]$ − 折扣因子根据即时和未来奖励的表现来决定。为了累积奖励的收敛，折扣因子应选择小于 1。在上述定义中，t 表示当前时间，而 T 是总时间，π 表示策略。目标是找到一种最优策略 π^* ，对于任何初始状态，都能最大化未来步骤的整体累积奖励，如下所示:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{T}\left(r\left(s_t, a_t\right) \mid \pi\right]\right.$$

By deriving the reward function and by maximizing the total rewards, the optimal solution can be reached.

通过推导奖励函数并最大化总奖励，可以找到最优解。

B. Markov Decision Process

B. 马尔可夫决策过程

Markov Decision Process (MDP) is a mathematically idealized framework used to help to make decisions on a stochastic environment. The next state of the system changes depending on the current state and action. MDPs have been studied and widely applied in many areas [22]. Recently, the problem of collision avoidance for UAVs is formulated as an MDP and solved by using different types of reinforcement algorithms. In [23], the collision avoidance problem for autonomous free flight of eVTOL vehicles was formulated as an MDP and solved by using Monte Carlo Tree Search algorithm. The result shows that the approach enables the eVTOL vehicles to reach their target destinations without any conflicts with other aircraft. The MDP methods can handle the collision avoidance problem online and offline. Because the policy is defined ahead of time, offline approaches are often not adaptable to changes in the environment. Online methods are more adaptive to changes in the environment because they can perceive new states and react appropriately, whereas the offline systems' polices are computed before take-off.

马尔可夫决策过程 (MDP) 是一个数学理想化的框架，用于帮助在随机环境中做出决策。系统的下一个状态取决于当前状态和动作。MDP 已经在许多领域进行了研究并得到了广泛应用 [22]。最近，无人机的碰撞避障问题被表述为 MDP，并通过使用不同类型的强化算法来解决。在 [23] 中，自主飞行 eVTOL 车辆的碰撞避障问题被表述为 MDP，并通过使用蒙特卡洛树搜索算法来解决。结果显示，这种方法使得 eVTOL 车辆能够无冲突地到达目标目的地。MDP 方法可以在线和离线处理碰撞避障问题。由于策略是提前定义的，离线方法通常无法适应环境的变化。在线方法更能适应环境的变化，因为它们能够感知新的状态并相应地做出反应，而离线系统的策略是在起飞前计算好的。

One main issue with the MDP algorithm is that they cannot handle large state spaces or continuous state spaces. As a result, it can be very expensive for large state spaces, and state space discretization can give mistakes. In order to overcome that problem, the deep reinforcement learning approach, which has advantage of both online and offline methods, has been recently used. Deep reinforcement learning methods can learn an approximation function to describe the policy over a continuous state-action space without having to discretize the state space though training. Recent growing performance of Deep Reinforcement Learning shows that a well-trained and well-designed AI agent is able to learn complicated strategies under uncertainty and then AI agent can provide a potential solution for UAM autonomous traffic control and management.

MDP 算法的一个主要问题是它们无法处理大型状态空间或连续状态空间。因此，对于大型状态空间来说，成本可能非常高，而状态空间离散化可能会导致错误。为了克服这个问题，最近使用了深度强化学习方法，该方法具有在线和离线方法的优势。深度强化学习方法可以通过训练学习一个近似函数来描述连续状态-动作空间上的策略，而无需对状态空间进行离散化。最近深度强化学习的性能提升表明，经过良好训练和设计的 AI 代理能够在不确定性下学习复杂的策略，因此 AI 代理可以为 UAM 自主交通控制和管理工作提供潜在的解决方案。

C. Policy-Based Learning

C. 基于策略的学习

Recently reinforcement learning has been investigated for conflict resolution and safe-separation assurance. The first research was introduce in [20] where the agent was trained and developed to deal with conflicts and minimize the delay of aircraft. Reinforcement learning has two basic algorithms: value-based and policy-based algorithms. In our work, we use the policy-based algorithm, which learns stochastic policies in model-free environment that has benefit of dealing with unknown environment and uncertainties in other agents' actions, unlike the value-based algorithms [25]. We are using the recent state-of-the-art policy-based algorithm, PPO (Proximal Policy Optimization), which is a recently developed policy-based technique that employs a neural network to approximate both the policy (actor) and the value function (critic) [26]. PPO is very easy to tune, has a good performance, and updates each step that minimizes the cost function in our work.

最近，强化学习已经被研究用于冲突解决和安全间隔保证。第一个研究在文献 [20] 中提出，其中代理被训练和开发来处理冲突并最小化飞机的延迟。强化学习有两种基本算法: 基于价值的算法和基于策略的算法。在我们的工作中，我们使用基于策略的算法，该算法在无需模型的环境中学习随机策略，这有利于处理未知环境和其他代理行为的确定性，与基于价值的算法 [25] 不同。我们使用的是最近最先进的基于策略的算法，ppo(近端策略优化)，这是一种最近开发的基于策略的技术，它使用神经网络来近似策略

(演员) 和价值函数 (评论家)[26]。PPO 非常容易调整，具有良好的性能，并且在我们的工作中每一步更新都能最小化成本函数。

PPO has become one of the most commonly used algorithms that is used to train robot hand to solve Rubik's cube or win games against professional players. It is basically a policy gradient algorithm that succeeded Trust Region Policy Optimization (TRPO) [26]. In our research, PPO has been used in order to create a more efficient and effective MARL model that focuses on off-policy approaches that store and reuse data for multiple policy updates rather than on-policy algorithms that use newly gathered training data before each update to agents' policies [27].

PPO 已成为最常用的算法之一，用于训练机器人手解魔方或战胜职业选手。它本质上是一种策略梯度算法，继承了信任域策略优化 (TRPO)[26] 的成功。在我们的研究中，PPO 被用于创建一个更高效、更有效的多智能体强化学习 (MARL) 模型，该模型专注于离策略方法，存储并重用数据以进行多次策略更新，而不是在每次更新智能体策略之前使用新收集的训练数据的策略算法 [27]。

D. Multi-Agent Reinforcement Learning

D. 多智能体强化学习

A multi-agent system is defined as a set of autonomous, interacting entities that share the same environment [28]. MARL involves multiple agents interacting with the same environment and each other while a single agent considers one agent's interaction with the environment. The environment, which is represented by a state vector, contains everything required for the agents to make a decision. When agents take an action, the environment is updated to a new state that depends on the behaviors of the environment. The agents are then rewarded based on the transition, which represents their goal in the environment that may be hidden form other agents. Figure 2 shows the process of an MARL problem and how the agents interact with the environment and the reward function structure for decision making. MARL has achieved great success on a wide range of multi-agent systems such as traffic light control, games, PowerGrid control, among others [29][30]. The main difficulty of MARL is that each agent has its own goal to achieve in the same environment, which may be unknown for other agents that create more complexity for agent's decision making, and in every additional agent the scale of the problem increases significantly [31]. Independent Q-learning is one of the most common algorithms to solve complex multi-agent reinforcement learning problems with high number of states. Independent Q-learning is a technique where each agent has its own action-observation system and its own network parameters and treats the other agents as a part of the environment [32]. This method may fail sometimes because when each agent learns and changes its own policy that affects the other agents' policy as well [33]. Independent Q-learning treats other agents as a part of environment without communication among agents thus can cause learning instability. In order to handle this learning instability, a group of agents need to be trained in a centralized manner with an open communication channel [34]. Communication channel is important for agents to interact successfully and solve the negotiation between agents.

多代理系统定义为一系列自治的、相互作用的实体，它们共享相同的环境 [28]。多代理强化学习 (MARL) 涉及多个代理在与相同环境和其他代理交互的同时，单个代理考虑一个代理与环境的交互。环境由状态向量表示，包含代理做出决策所需的一切。当代理采取行动时，环境会更新到一个新的状态，该状态取决于环境的各种行为。然后根据这种转换，代理会获得奖励，这种转换代表了代理在环境中的目标，可能对其他代理隐藏。图 2 展示了 MARL 问题的过程以及代理如何与环境交互和决策的奖励函数结构。MARL 在广泛的多元代理系统中取得了巨大成功，如交通信号控制、游戏、PowerGrid 控制等 [29][30]。MARL 的主要困难在于，每个代理都在相同的环境中实现自己的目标，这些目标对其他代理可能是未知的，从而为代理的决策增加了更多的复杂性，而且每个新增的代理都会使问题的规模显著增加 [31]。独立 Q 学习是解决具有高状态数量的复杂多代理强化学习问题最常见的方法之一。独立 Q 学习是一种技术，其中每个代理都有自己的动作-观察系统和自己的网络参数，并将其他代理视为环境的一部分 [32]。这种方法有时可能会失败，因为当每个代理学习和改变自己的策略时，也会影响其他代理的策略 [33]。独立 Q 学习将其他代理视为环境的一部分，而不考虑代理之间的通信，因此可能导致学习不稳定。为了处理这种学习不稳定性，需要以集中方式训练一组代理，并开放通信通道 [34]。通信通道对于代理成功交互和解决代理之间的协商至关重要。

Centralized learning decentralized execution, an alternative and one of the promising approaches to independent Q-learning, is used in our research to handle learning instability. With this approach, a group of agents can be trained at the same time by applying a centralized approach to shared learning experiences [34]. The goal of the centralized learning with decentralized execution paradigm is to allow agents access to information that normally would be restricted during execution. During training, agents

can view other agents' observation and actions in order to get a more complete picture of the global state of the system (instead of receiving only a local observation based on its own pair observation and state). In this way, agents learn policies that take into consideration the needs of others and can be applied at execution time in a decentralized way. Learning efficiency can be improved via centralized learning that depends on the experiences of all agents. With decentralized policies each agent can take their local actions based on their observation and in limited communications during execution [35]. Centralized learning with decentralized execution has become a typical solution to multi-agent systems, where agents are trained offline using centralized information but execute in a decentralized manner online.

集中式学习分布式执行，作为一种替代方法，也是独立 Q 学习的一种有前景的途径，在我们的研究中用于处理学习不稳定性。采用这种方法，一组代理可以同时通过应用集中式方法到共享的学习经验来进行训练 [34]。集中式学习与分布式执行范式的目标是让代理能够访问在执行过程中通常受限的信息。在训练期间，代理可以查看其他代理的观察和行动，以获得系统的全局状态的更完整图像 (而不是仅接收到基于自身的观察和状态的本地观察)。这样，代理学习到的策略会考虑到其他代理的需求，并且可以在执行时以分布式方式应用。通过依赖所有代理的经验的集中式学习，可以提高学习效率。在分布式策略下，每个代理可以在执行期间基于他们的观察和有限的通信采取本地行动 [35]。集中式学习与分布式执行已成为多代理系统的典型解决方案，其中代理在离线时使用集中式信息进行训练，但在在线时以分布式方式执行。

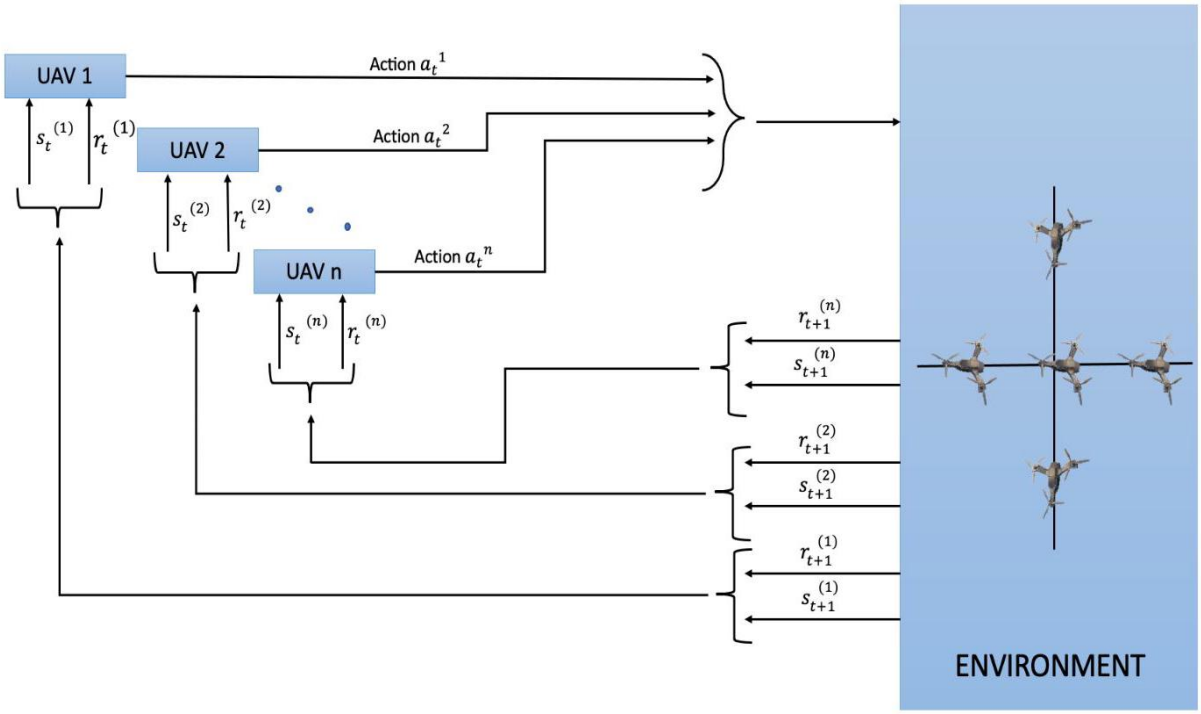

Fig. 2 Multi-agent reinforcement learning scheme for vehicle coordination at UAM intersection.
图 2 多代理强化学习方案，用于城市空中交通 (UAM) 交叉口的车辆协调。

III. System Models and Problem Formulation

III. 系统模型与问题构建

This research focuses on the applicability of reinforcement learning for UAM intersection scenarios for safe separation between eVTOL vehicles at the intersections and during the flight. To build a reinforcement learning model for autonomous air traffic management at an intersection, we need to define the states, actions, and rewards. To validate our model, we used the BlueSky simulator environment [13]. This simulation software is not built for eVTOL vehicles, but it provides real aircraft data, which allows us to run our model in a real-world environment. We develop one simple intersection case study to test our MARL model. The BlueSky simulator has so many different types of aircrafts. In BlueSky simulation we choose all the aircraft as the same type, Airbus A318. We assume that all aircraft maintain their

route during the flight. Figure 3 below shows our single intersection case in BlueSky air traffic control environment. The vertical and horizontal white lines represent the example air corridors, the green triangle shapes represent each aircraft along the route, and the yellow square shapes symbolize airports on the map. One single intersection has been studied and it can be applied to multi-intersection cases as well.

本研究关注强化学习在 UAM 交叉场景中的应用，以确保在交叉点和飞行过程中 eVTOL 车辆之间的安全间隔。为了构建一个用于交叉点自主空中交通管理的强化学习模型，我们需要定义状态、动作和奖励。为了验证我们的模型，我们使用了 BlueSky 模拟器环境 [13]。这个模拟软件并非专为 eVTOL 车辆设计，但它提供了真实飞机数据，这让我们得以在现实世界环境中运行我们的模型。我们开发了一个简单的交叉案例研究来测试我们的 MARL 模型。BlueSky 模拟器中有许多不同类型的飞机。在 BlueSky 模拟中，我们选择所有飞机为同一类型，即空客 A318。我们假设所有飞机在飞行过程中保持其航线。下图 3 展示了我们在 BlueSky 空中交通控制环境中的单个交叉案例。垂直和水平的白色线条代表示例空中走廊，绿色三角形代表每个飞机的航线，黄色正方形形状象征着地图上的机场。已经研究了一个单独的交叉点，并且它可以应用于多交叉点的情况。

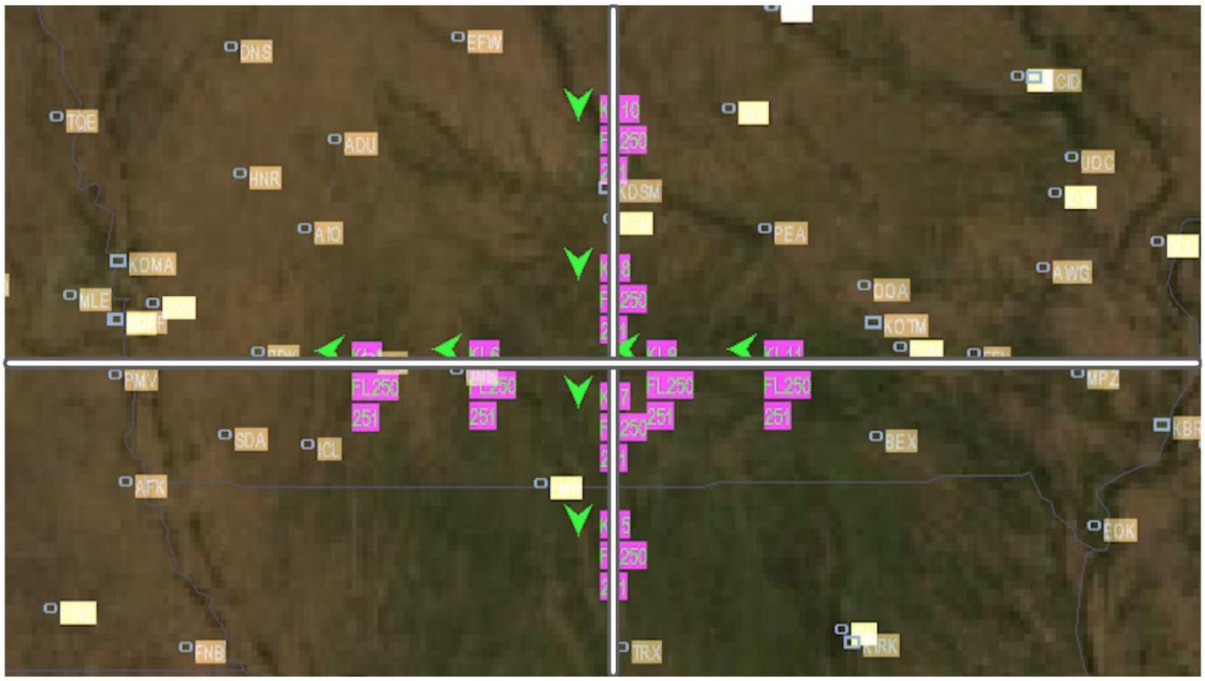

Fig. 3 Single intersection case study in BlueSky environment.
图 3 BlueSky 环境中的单个交叉案例研究。

A. Problem Formulation

A. 问题构建

An MARL involves multiple agents interacting with the same environment and each other while a single agent considers one agent's interaction with environment, as we explained in the previous section. In our research, we use multi-agent system to gain a better understanding of the behavior of each eVTOL vehicle during the flight and at the intersection.

一个 MARL 涉及多个代理在与相同环境和其他代理交互，而单个代理则考虑一个代理与环境的交互，正如我们在上一节中解释的那样。在我们的研究中，我们使用多代理系统来更好地理解每个 eVTOL 车辆在飞行和交叉过程中的行为。

Our problem is formulated as an MARL problem by assuming each eVTOL vehicle as an agent and developing a model algorithm that runs on each vehicle to provide speed guidance in order to help each eVTOL vehicle to reach its destination safely and on time by avoiding the long waiting time and potential conflicts at intersections. We will define the states, actions, and rewards for agents as follows.

我们将问题构建为一个 MARL 问题，假设每个 eVTOL 车辆为一个代理，并开发一个在每辆车上运行的模型算法，以提供速度指导，帮助每个 eVTOL 车辆通过避免交叉点的长时间等待和潜在的冲突，安全准时地到达目的地。我们将如下定义代理的状态、动作和奖励。

B. State Space

B. 状态空间

Each individual aircraft makes decisions based on the information stored in the state. The state space is where the agent gets information to make decisions. To make our problem more visible, we assume that each aircraft position and dynamics are available continuously. We use the LSTM network to store the intruder's information as a fixed-length vector which allows the agent to reach the all-intruder information with a large number of agents in the environment. First, LSTM processes the intruder information that includes the speed, acceleration, distance to the goal, distance to the intersection, and distance between intruders, and turns that information into a fixed-length vector. Then, LSTM network encodes the intruder information and is trained to learn for ownship to decide which intruder needs to be considered.

每架飞机根据存储在状态中的信息做出决策。状态空间是代理获取决策信息的场所。为了使我们的问题更加直观，我们假设每架飞机的位置和动力学信息是连续可用的。我们使用 LSTM 网络将入侵者的信息存储为固定长度的向量，这使得代理能够在环境中与大量代理交互时获取所有入侵者的信息。首先，LSTM 处理包含速度、加速度、到目标的距离、到交叉点的距离以及入侵者之间的距离的入侵者信息，并将这些信息转换为固定长度的向量。然后，LSTM 网络对入侵者信息进行编码，并训练学习以使自有机决定需要考虑哪个入侵者。

UAM corridors are environed as safe and efficient mechanisms for high-density air traffic missions. Figure 4 below shows our single intersection scenario where each eVTOL has its own destination. We assumed that the traffic density in main and secondary air corridors are the same. Each eVTOL has to pass the intersection without causing any conflict during the flight and when passing the intersection. Since our problem is a single-lane intersection scenario, we can only focus on single way traffic flow.

UAM 走廊被视为安全且高效的高密度航空任务机制。下图 4 展示了我们的单一交叉场景，其中每个 eVTOL 有其自己的目的地。我们假设主次航空走廊的交通密度相同。每个 eVTOL 在飞行过程中和通过交叉点时都不能造成任何冲突。由于我们的问题是单车道交叉场景，因此我们只需关注单向车流。

We use a similar approach as in [20] for formulating the state and intruder state information for the agents:

我们采用与文献 [20] 类似的方法来制定代理的状态和入侵者状态信息:

$$S_t^0 = \left(d^0{}_{\text{goal}}, v^0, a^0, r^0, d^{LOS}\right)$$

$$h_t^0(i) = \left(d^i{}_{\text{goal}}, v^i, a^i, r^i, d_0^i, d^0{}_{\text{int}}, d^i{}_{\text{int}}\right)$$

where S_t^0 represents the state of the ownship, and $h_t^0(i)$ represents the information of the i th intruder that is available at time t .

其中 S_t^0 表示自机的状态，$h_t^0(i)$ 表示在时间 t 可用的第 i 个入侵者的信息。

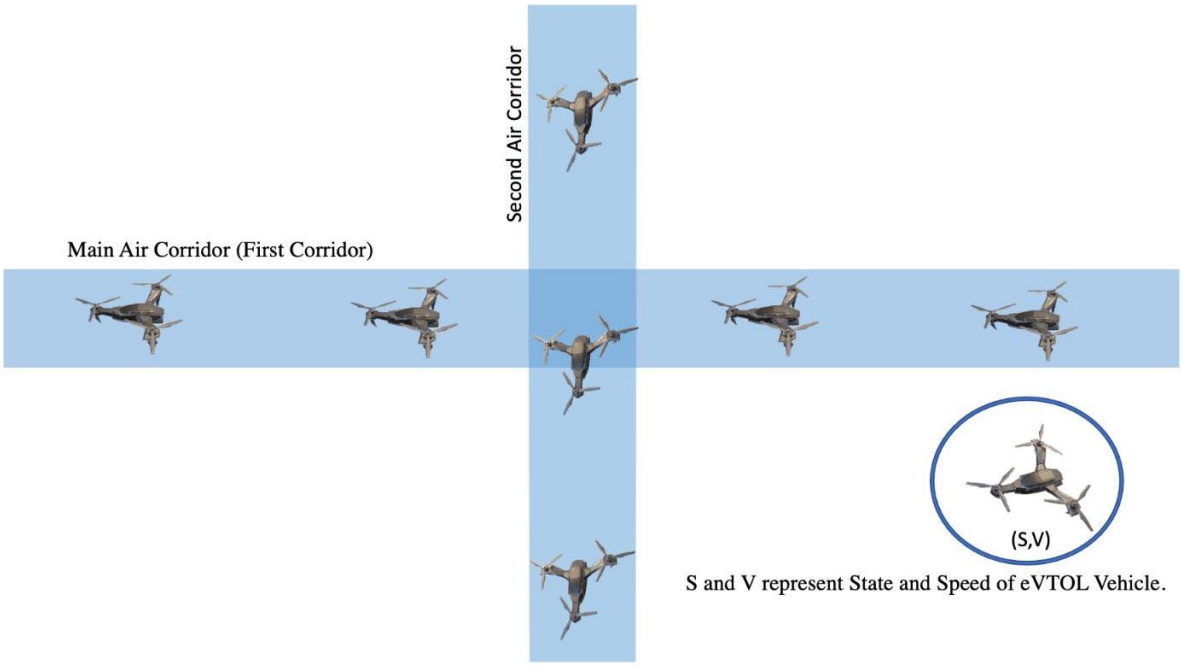

Fig. 4 Illustration of an UAM intersection scenario.
图 4 UAM 交叉场景的示意图。

C. Action Space

C. 动作空间

Our model needs to take an appropriate action to guide the aircraft well at the intersection by providing speed advisories. There are three actions that any aircraft on the same route can take: accelerate, decelerate, and maintain the current speed or no acceleration. In order to limit the number of actions that each agent takes, we limit each agent to choose an action every 5 seconds. This limitation is sufficient for safe separation of the vehicles considered as a case study in this paper. The action space can be defined as follows:

我们的模型需要在交叉点处通过提供速度建议来正确引导飞机。在任何相同航线上的飞机可以采取三种行动: 加速、减速和保持当前速度或不加速。为了限制每个代理采取的行动数量，我们限制每个代理每 5 秒选择一次行动。这个限制对于本文案例研究中考虑的车辆安全间隔是足够的。行动空间可以定义如下:

$$A_t = [V_-, 0, V_+]$$

V_- : decelerate.
V_- : 减速。
0 : no acceleration or maintain current speed.
0 : 无加速或保持当前速度。
V_+ : accelerate.
V_+ : 加速。

D. Reward Function

D. 奖励函数

We formulate identical reward functions for all the agents. Each agent tries to gain more reward locally and if two agents are in conflict, they will receive a penalty. The conflict is defined when the distance

between two eVTOL vehicles is less than 2 nautical miles $\left(d^{LOS}=2\right)$. We use the reward function as follows:

我们为所有代理制定了相同的奖励函数。每个代理都试图局部获得更多奖励，如果两个代理发生冲突，它们将会受到惩罚。当两辆 eVTOL 车辆之间的距离小于 2 海里 $\left(d^{LOS}=2\right)$ 时，我们定义发生了冲突。我们使用以下奖励函数:

$$G_t = R_t + \gamma R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$$R_t = \begin{cases} -1 & \text{if } d^{LOS} < 2NM \\ \varepsilon & \text{if } 2 \le d^{LOS} \le 4 \\ 0 & \text{else} \end{cases}$$

G_t : total return.

G_t ：总回报。

γ : discount factor.

γ ：折扣因子。

ε : some small negative reward when the vehicle gets to the conflict distance.

ε : 当车辆达到冲突距离时的一些小的负奖励。

d_0^c : the distance from the ownship to the closest eVTOL vehicle.

d_0^c : 从本机到最近的 eVTOL 车辆的距离。

α, δ : small positive constants to penalize agents.

α, δ : 小的正常数，用于惩罚代理。

where γ is a parameter called discount factor and $0 \le \gamma \le 1$. The discount factor determines the present value of future rewards: a reward received k time steps in the future is worth only γ^{k-1} times what it would be worth if it were received immediately. The agent will be trained to learn and gain more rewards based on our model.

其中 γ 是称为折扣因子的参数，$0 \le \gamma \le 1$ 。折扣因子决定了未来奖励的现值: 在未来 k 时间步获得的奖励，其价值仅相当于立即获得时的 γ^{k-1} 倍。代理人将根据我们的模型进行训练，以学会获取更多奖励。

The terminal state is where all the agents reach their target positions. The goal of the MARL model is to find the optimal policy for the agents to reach their terminal states without any conflict.

终止状态是指所有代理人都达到目标位置的状态。多智能体强化学习模型的目标是找到代理人不发生冲突地达到终止状态的最优策略。

The terminal state reward structure is as follows:

终止状态的奖励结构如下:

- + 20 when all the vehicles arrived at their target positions.
- 当所有车辆到达目标位置时，+20。
- - 5 when a vehicle did not arrive at its target position.
- 当一辆车未到达目标位置时，-5。
- -10 when vehicles collided along the way.
- 当车辆在途中相撞时，-10。

The task of each agent is to maximize its long-term performance and receive the long-term rewards, while only receiving feedback about one-step performance. By computing an optimal action-value function, we can find the optimal policy by updating our value function after each episode run. The action-value function, Q , will converge to the optimal, Q^* , action-value function independently for each agent after the model trained. We use Independent Q-learning where agents learn independently and communicate through a communication channel for negotiation when they have conflicts and instability.

每个代理人的任务是最大化其长期性能并获得长期奖励，而仅获得单步性能的反馈。通过计算最优动作价值函数，我们可以在每个回合结束后更新我们的价值函数，从而找到最优策略。动作价值函数 Q 将在模型训练后独立地收敛于每个代理人的最优 Q^* 动作价值函数。我们使用独立 Q 学习，其中代理人独立学习，并在冲突和不稳定时通过通信通道进行协商。

The action-value function (Q-learning) for estimating optimal policy is as follows:

估计最优策略的动作价值函数 (Q 学习) 如下:

$$Q(S,A) \leftarrow Q(S,A) + \delta\left[R + \gamma \max_{a} Q(S',a) - Q(S,A)\right]$$

where S represents the current state, A is the action chosen in that state, $\delta \in (0,1]$ is the learning rate, and S' is the next state of the agent. We loop the action-value function for each step episode until converging to the optimal action-value function that provides the optimal policy.

其中 S 表示当前状态，A 是在该状态下选择的行为，$\delta \in (0,1]$ 是学习率，S' 是代理人的下一个状态。我们循环每个步骤的动作价值函数，直到收敛到提供最优策略的最优动作价值函数。

IV. Problem Solution and Case Study

IV. 问题解决方案与案例研究

In the previous section, we formulated the vehicle coordination problem at an UAM intersection as an MARL problem and defined the reward structure of the problem that leads the agent to find the optimal policy. In this section, we develop the solution to the formulated problem by using neural networks. We use one neural network to train and test our MARL model to distribute the best possible speed advisories to each agent without causing any conflict during the flight and at the intersection. In order to formulate the MARL problem, we implement a centralized learning, decentralized execution (CLDE) approach using one neural network that is shared by all agents. We train the MARL model that is cooperated with all the agents through that network. While the policy-based MARL model is formed as CLDE that helps the agent to learn their local policies [36] and all the agents share the same neural network, their actions still can be different in the execution. Figure 5 below shows the MARL neural network architecture. This neural network can be implemented to all individual eVTOL vehicles, instead of having a specific neural network for each eVTOL vehicle for safe separation at the intersection and during the flight.

在上一节中，我们将 UAM 交叉口的车辆协同问题公式化为多智能体强化学习 (MARL) 问题，并定义了问题的奖励结构，引导智能体找到最优策略。在本节中，我们通过使用神经网络来开发所提出问题的解决方案。我们使用一个神经网络来训练和测试我们的 MARL 模型，以便在飞行和交叉口期间向每个智能体分配最佳的速度建议，而不会造成任何冲突。为了公式化 MARL 问题，我们实施了一个集中学习、分布式执行的 (CLDE) 方法，所有智能体共享一个神经网络。我们训练了一个与所有智能体合作的 MARL 模型。虽然基于策略的 MARL 模型形成 CLDE，帮助智能体学习他们的局部策略 [36]，并且所有智能体共享同一个神经网络，但他们在执行时的行动仍然可以不同。下图 5 展示了 MARL 神经网络的架构。这个神经网络可以被实施到所有的单个 eVTOL 车辆上，而不是为每个 eVTOL 车辆在交叉口和飞行中安全分离配备特定的神经网络。

The first layer of the network is an LSTM where all the intruder information is encoded as a fixed-length vector. Then, the information is sent to two fully connected layers to execute from the output for the best policy and value function, and then each eVTOL vehicle follows this policy until termination. Since the environment is stochastic, the policy on one episode is not clear to decision making for each agent. Therefore, by collecting multiple episodes we can update the neural network policy where the network can execute different outcomes from the same policy.

网络的第一层是一个 LSTM，所有入侵者信息都被编码为一个固定长度的向量。然后，这些信息被发送到两个全连接层，以执行输出最佳策略和价值函数，之后每个 eVTOL 车辆遵循这个策略直到终止。由于环境是随机的，一个剧集中的策略对于每个智能体的决策并不明确。因此，通过收集多个剧集，我们可以更新神经网络的策略，网络可以从相同的策略中执行不同的结果。

The reinforce algorithm we used is a policy-based algorithm, PPO, which is an optimization for our network to share layer between the actor and the critic. PPO is one of the most popular on-policy RL algorithms but it is less utilized than off-policy algorithms [37]. In this network, PPO is used with two identical layers with 128 nodes and LSTM encoder with 32 nodes. In order to activate the neurons, we used the rectified linear activation function or known as RELU activation function for both the hidden layers and the LSTM layer in our neural networks. RELU function is one of the default activation functions for neural networks due to its performance and ease of training. Softmax activation function is used for the first input layer, and for the last input the linear activation function is used. Softmax activation function scales numbers into probabilities and often uses as the last activation function of neural networks [38]. At the end of the network, two fully connected layers (hidden layers) produce the policy and the value for the given state. The policy is distributed at the beginning of each episode and updated at the end of each episode.

我们使用的强化算法是一种基于策略的算法，即 PPO(Proximal Policy Optimization)，它优化了我们的网络，使演员 (actor) 和评论家 (critic) 之间共享层。PPO 是最流行的同策略 (on-policy) 强化学习算法之一，但相较于异策略 (off-policy) 算法的使用频率较低 [37]。在这个网络中，PPO 与两个具有 128 个节点的相同层和具有 32 个节点的 LSTM 编码器一起使用。为了激活神经元，我们在神经网络的隐藏层和 LSTM 层中使用了修正线性激活函数，也被称为 RELU 激活函数。RELU 函数由于其性能和易于训练的特点，成为神经网络的默认激活函数之一。Softmax 激活函数用于第一个输入层，而对于最后一个输入，则使用线性激活函数。Softmax 激活函数将数值缩放到概率范围，并常作为神经网络的最后一个激活函数 [38]。在网络末端，两个全连接层 (隐藏层) 为给定的状态生成策略和价值。策略在每个剧集的开始时分配，并在每个剧集结束时更新。

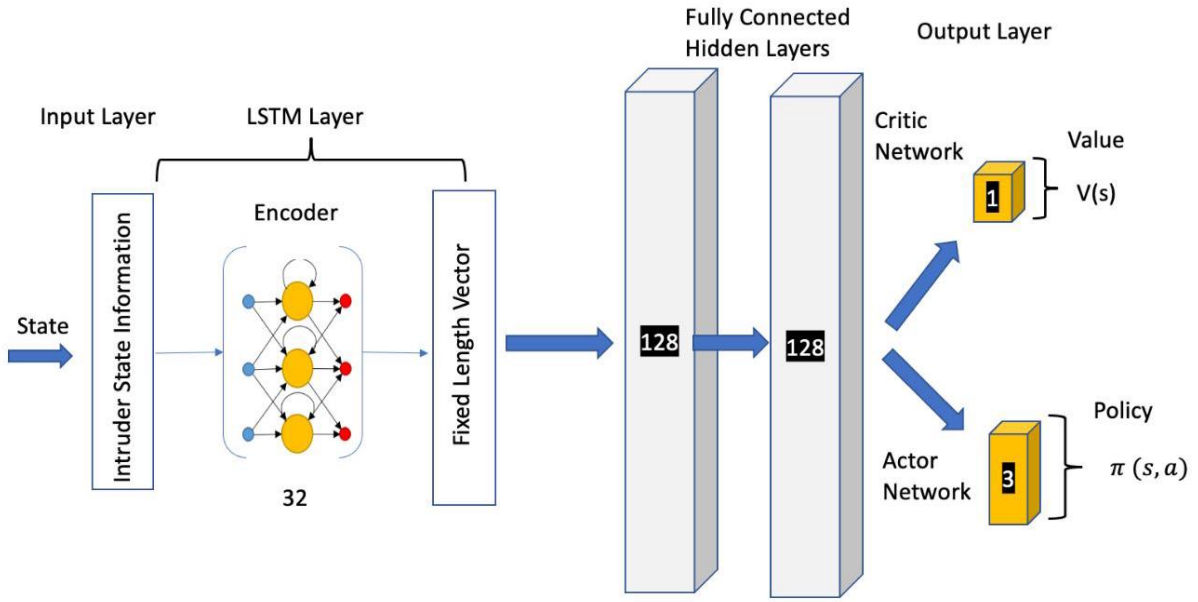

Fig. 5 MARL neural network architecture.

图 5 多智能体强化学习 (MARL) 神经网络架构。

In Figure 5, we can see that the information of each eVTOL vehicle is pre-processed through the input layer and goes to the LSTM layer. LSTM reads the inputs data and encodes it into a fixed-length numeric vector. Then, the output information goes through two fully connected layers. The policy network is called the actor since it selects the action to take and the value function network is called the critic since it evaluates how good the action is. The actor and critic network share the same layers of the network. The policy and value are derived from the fully connected output layer of 4 nodes at the end of our network. With this neural network, we can update the policy of each agent at the beginning of each episode. Each eVTOL vehicle follows its own policy until it reaches its destination.

在图 5 中，我们可以看到每辆 eVTOL 车辆的信息通过输入层预处理后传递到 LSTM 层。LSTM 读取输入数据并将其编码为固定长度的数值向量。然后，输出信息通过两个全连接层。策略网络被称为演员，因为它选择要执行的动作；价值函数网络被称为评论家，因为它评估动作的好坏。演员和评论家网络共享网络的相同层。策略和价值是从我们网络末端的 4 个节点的全连接输出层派生出来的。有了这个神经网络，我们可以在每个情节开始时更新每个代理的策略。每辆 eVTOL 车辆遵循自己的策略，直到到达目的地。

After all parameters are tuned and the model runs almost 20.000 episodes, the model is confidently working and learning different scenarios in the environment. With that single network, each eVTOL vehicle can decide its own speed for safe separation. The main goal is accomplished when each vehicle reaches their destination without any conflict during the travel time and at intersection. In our case study, we only focus on a single intersection to develop our MARL model. With the same approach, multi-intersection problems could be handled as well.

在所有参数调整完毕且模型运行了近 20,000 个情节之后，模型可以自信地工作并在环境中学习不同的场景。有了这个单一网络，每辆 eVTOL 车辆可以决定自己的速度以确保安全间隔。主要目标是在每个车辆在旅行时间和交叉路口没有冲突的情况下到达目的地时实现。在我们的案例研究中，我们只关注一个交叉路口来开发我们的 MARL 模型。使用同样的方法，也可以处理多交叉路口问题。

V. Model Performance

V. 模型性能

We used BlueSky air traffic control simulator to demonstrate the performance of our MARL model with real-world simulation experiment. BlueSky is an open-source and open-data multi-platform simulation tool [13]. BlueSky is an example of real-world flight ATC simulators that has recently been used to demonstrate reinforcement learning algorithms and test MARL model performance. Since the BlueSky simulation environment does not provide an eVTOL vehicle, we tested our MARL model with the fixed-wing Airbus A318 aircraft. When we run the simulations, we assume that the initial position and initial speed of each aircraft do not change in every episode run. Each simulation run in the BlueSky is referred to as one episode. During our case study, each episode consisted of 12 aircraft randomly entering the airspace for model training. Each aircraft has its own goal to accomplish in the environment. We ran our model 3,500 episode to train agents and evaluated the model performance. At the end of training, we found that each agent can successfully reach their destination and pass the intersection without causing conflicts. During each episode, each aircraft's initial position, initial speed, and destination are randomly generated in the BlueSky environment. Training and testing our model with high number of small UAVs will be giving more accurate results for UAM operations. Due to available aircraft in BlueSky environment, we were only able to test our model with Airbus A318 which is the smallest aircraft in the BlueSky environment. From Figure 6 that shows the results of 3,500 episode runs, we can see that the model behavior gets better over time. While the red curve represents the number of collisions, the green curve shows the success pass at the intersection. The blue curve shows the mean success of the model over episodes, and the orange curve shows the mean collision of the model per episode run. After 2,500 episode runs, our model has a better learning performance and generates better policies that lead each agent for safe separation at the intersection.

我们使用 BlueSky 空中交通管制模拟器来展示我们的 MARL 模型在实际世界模拟实验中的性能。BlueSky 是一款开源且开放数据的多平台模拟工具 [13]。BlueSky 是现实世界飞行 ATC 模拟器的一个例子，最近已被用于展示强化学习算法并测试 MARL 模型的性能。由于 BlueSky 模拟环境没有提供 eVTOL 飞行器，我们使用固定翼的空客 A318 飞机测试了我们的 MARL 模型。当我们运行模拟时，我们假设每个飞机的初始位置和初始速度在每次剧集运行中都不会改变。BlueSky 中的每次模拟运行被称为一个剧集。在我们的案例研究中，每个剧集由 12 架飞机随机进入空域进行模型训练。每架飞机在环境中都有自己的目标要完成。我们运行了 3500 个剧集来训练智能体并评估模型性能。训练结束时，我们发现每个智能体都能成功到达目的地并在交叉点通过而不造成冲突。在每集期间，每架飞机的初始位置、初始速度和目的地都是在 BlueSky 环境中随机生成的。用大量小型无人机训练和测试我们的模型将为 UAM 操作提供更准确的结果。由于 BlueSky 环境中可用的飞机，我们只能用 BlueSky 环境中最小的飞机空客 A318 来测试我们的模型。从图 6 中显示 3500 个剧集运行结果的图表中，我们可以看到模型的行为随着时间的推移变得更好。红色曲线代表碰撞数量，绿色曲线显示在交叉点的成功通过次数。蓝色曲线显示了模型在剧集中的平均成功率，橙色曲线显示了每集运行中的平均碰撞次数。在 2500 个剧集运行后，我们的模型具有更好的学习性能并生成更好的策略，引导每个智能体在交叉点安全分离。

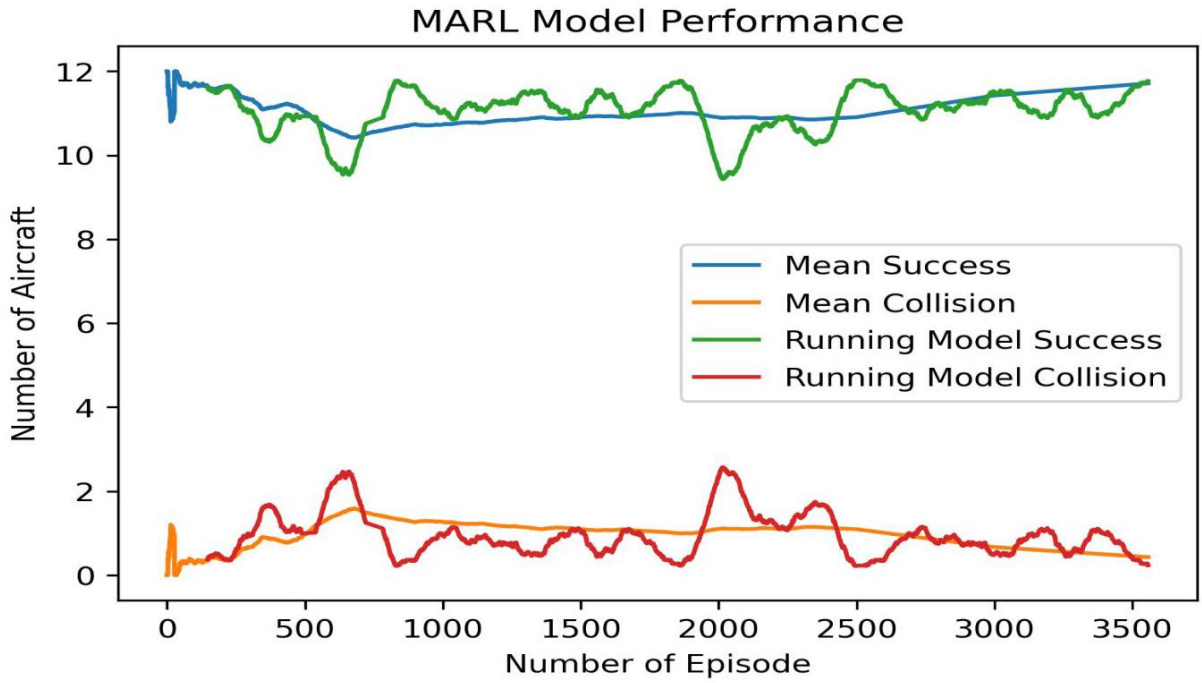

Fig. 6 MARL model performance for single intersection case study.
图 6 多智能体强化学习模型在单一交叉口案例研究中的性能。

VI. Conclusion
VI. 结论

With recent development in AI and deep reinforcement learning, we are able to formulate the traffic control problems including intersection control for both ground and air vehicles to enable intelligent transportation systems. Deep reinforcement learning is one of the promising ways to handle vehicle coordination and management problem at intersections for both ground vehicles and air vehicles. In this work, we showed the application of multi-agent reinforcement learning (MARL) techniques on air traffic control problems. A recent LSTM architecture is used to encode the intruder information into a fixed length vector and then execute that information to the next layer.

随着人工智能和深度强化学习的最新发展，我们能够将包括地面和空中交通工具在内的交通控制问题 (包括交叉口控制) 公式化为实现智能交通系统的形式。深度强化学习是处理地面和空中交通工具在交叉口进行车辆协调和管理问题的一种有前景的方法。在这项工作中，我们展示了多智能体强化学习 (MARL) 技术在空中交通控制问题上的应用。最近的长短时记忆 (LSTM) 架构被用来将入侵者信息编码成一个固定长度的向量，然后将该信息传递到下一层。

Intersection control is one of the recent problems that has been raising in UAM. In our research, we developed an MARL model to solve a single intersection problem for UAM. The problem is formulated as MARL and solved by using MARL neural network for training and testing. After training and testing, the aircraft (i.e., agent) can avoid conflicts during the travel time. Our model is tested in the BlueSky simulation environment to visualize the result and check the final performance. In our future work, we will develop our model to solve regional UAM problems [39] inside the Air Corridors, as proposed by the FAA and NASA [5]. The availability of Corridors will be determined by Air Traffic Control (ATC). The future work will be on developing the MARL model to handle the intersections within the entire UAM Corridor with more complex scenarios for safe separation to fully control the air traffic autonomously with high efficiency.

交叉口控制是最近在城市化空中交通 (UAM) 中提出的一个问题。在我们的研究中，我们开发了一个 MARL 模型来解决 UAM 中的单一交叉口问题。问题被公式化为 MARL，并通过使用 MARL 神经网络进行训练和测试来解决。训练和测试后，飞机 (即智能体) 可以在飞行时间内避免冲突。我们的模型在 BlueSky 仿真环境中进行了测试，以可视化结果并检查最终性能。在未来的工作中，我们将开发我们的模型来解决区域性的 UAM 问题 [39]，即在 FAA 和 NASA[5] 提出的空中走廊内部的问题。走廊的可用性将由空中交通管制 (ATC) 确定。未来的工作将是开发 MARL 模型，以处理整个 UAM 走廊内的交叉口，在更复杂的场景中进行安全分离，以高效率地自主控制空中交通。

References

参考文献

[1] D. Sources, "Office of Aviation Policy and Plans (APO-100) FAA U . S . Passenger Airline Forecasts, Fiscal Years 2020-2040," pp. 1-66, 2020.

[1] D. Sources, "Office of Aviation Policy and Plans (APO-100) FAA U.S. Passenger Airline Forecasts, Fiscal Years 2020-2040," pp. 1-66, 2020.

[2] FAA, "Concept of Operations v2.0," Enabling Civ. Low-altitude Airsp. Unmanned Aircr. Syst. Oper., p. https://utm.arc.nasa.gov/index.shtml, 2020.

[2] FAA, " 操作概念 v2.0", 民用低空空域无人机系统运行启用 ,p. https://utm.arc.nasa.gov/index.shtml, 2020.

[3] M. Johnson et al., "Flight test evaluation of an unmanned aircraft system traffic management (UTM) concept for multiple beyond-visual-line-of-sight operations," 12th USA/Europe Air Traffic Manag. R D Semin., no. June, 2017.

[3] M. Johnson 等人, " 对多架超视距无人机系统交通管理 (UTM) 概念的飞行测试评估", 第 12 届美国/欧洲空中交通管理研发研讨会, 编号 June, 2017.

[4] A. S. Aweiss, B. D. Owens, J. L. Rios, J. R. Homola, and C. P. Mohlenbrink, "Unmanned Aircraft Systems (UAS) Traffic Management (UTM) National Campaign II," AIAA Inf. Syst. Infotech Aerospace, 2018, no. 209989, pp. 1-16, 2018.

[4] A. S. Aweiss, B. D. Owens, J. L. Rios, J. R. Homola 和 C. P. Mohlenbrink, " 无人机系统 (UAS) 交通管理 (UTM) 国家活动 II", AIAA 信息系统信息科技航空航天, 2018, 编号 209989, pp. 1-16, 2018.

[5] S. Bradford, "Urban Air Mobility (UAM) Concept of Operations v1.0," pp. 1-42, 2020.

[5] S. Bradford, " 城市空中出行 (UAM) 操作概念 v1.0", pp. 1-42, 2020.

[6] I. Greenfeld, "Concept of Operations for Urban Air Mobility Command and Control Communications," no. April, 2019.

[6] I. Greenfeld, " 城市空中出行指挥与控制通信操作概念", 编号 April, 2019.

[7] S. Hasan, "Urban Air Mobility (UAM) Market Study," Demand Emerg. Transp. Syst., no. November 2018, pp. 267-284, 2019.

[7] S. Hasan, " 城市空中出行 (UAM) 市场研究", 需求出现的交通系统, 编号 November 2018, pp. 267-284, 2019.

[8] J. Holden and N. Goel, "Fast-Forwarding to a Future of On-Demand Urban Air Transportation," pp. 1-98, 2016.

[8] J. Holden 和 N. Goel, " 快速前进到按需城市空中交通的未来", pp. 1-98, 2016.

[9] Airbus, "Urban Air Mobility by Airbus," vol. 33, no. 0, pp. 1-3, 2018.

[9] 空中客车公司, " 空中客车公司的城市空中出行", 卷 33, 编号 0, pp. 1-3, 2018.

[10] Corgan, "Connect Evolved," 2019.

[10] Corgan, "Connect Evolved," 2019.

[11] Google, "AlphaGo | DeepMind," Google, 2018. .

[11] Google, "AlphaGo | DeepMind," Google, 2018.

[12] OpenAI, "OpenAI," OpanAI, 2019. .

[12] OpenAI, "OpenAI," OpanAI, 2019.

[13] J. Hoekstra, J. Ellerbroek, and J. M. Hoekstra, "BlueSky ATC Simulator Project: an Open Data and Open Source Approach Three-Dimensional Airborne Separation Assistance Displays View project BlueSky-Open source ATM simulator View project BlueSky ATC Simulator Project: an Open Data and Open Source Approach," seventh Int. Conf. Res. Air Transp., no. June, 2016.

[13] J. Hoekstra, J. Ellerbroek, 和 J. M. Hoekstra, "BlueSky ATC Simulator Project: an Open Data and Open Source Approach Three-Dimensional Airborne Separation Assistance Displays View project BlueSky-Open source ATM simulator View project BlueSky ATC Simulator Project: an Open Data and Open Source Approach," 第七届国际航空运输研究会议, 第 June 卷, 2016 年。

[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol. 9, no. 8, pp. 1735-1780, 1997.

[14] S. Hochreiter 和 J. Schmidhuber, " 长短期记忆," 神经计算, 第 9 卷, 第 8 期, pp. 1735-1780, 1997 年。

[15] X. Liang, X. Du, G. Wang, and Z. Han, "Deep reinforcement learning for traffic light control in vehicular networks," arXiv, vol. XX, no. Xx, pp. 1-11, 2018.

[15] X. Liang, X. Du, G. Wang, 和 Z. Han, " 车辆网络中交通灯控制的深度强化学习," arXiv, 第 XX 卷, 第 Xx 期, pp. 1-11, 2018 年。

[16] W. Genders and S. Razavi, "Using a Deep Reinforcement Learning Agent for Traffic Signal Control," pp. 1-9, 2016.

[16] W. Genders 和 S. Razavi, " 使用深度强化学习代理进行交通信号控制," pp. 1-9，2016 年。

[17] H. Erzberger, "Automated Conflict Resolution for Air," 25Th Int. Congr. Aeronaut. Sci., no. March, pp. 1-28, 2014.

[17] H. Erzberger, "Automated Conflict Resolution for Air," 第 25 届国际航空科学大会，第 March 卷，pp. 1-28，2014 年。

[18] H. Erzberger and K. Heere, "Algorithm and operational concept for resolving short-range

[18] H. Erzberger 和 K. Heere, " 解决短程冲突的算法和操作概念"

conflicts," Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng., vol. 224, no. 2, pp. 225-243, 2010.

[19] K. Tumer and A. Agogino, "Adaptive management of air traffic flow: A multiagent coordination approach," Proc. Natl. Conf. Artif. Intell., vol. 3, pp. 1581-1584, 2008.

[20] M. Brittain, X. Yang, and P. Wei, "A Deep Multi-Agent Reinforcement Learning Approach to Autonomous Separation Assurance," arXiv, pp. 1-26, 2020.

[21] Uber Elevate, "Operations Inside Corridors," no. October, 2020.

[22] "Markovian decision processes," Mathematics in Science and Engineering, vol. 130, no. C. pp. 172-187, 1977.

[23] X. Yang, L. Deng, and P. Wei, "Multi-agent autonomous on-demand free flight operations in urban air mobility," AIAA Aviat. 2019 Forum, no. June, pp. 1-13, 2019.

[24] A. Zanette, M. J. Wainwright, and E. Brunskill, "Provable Benefits of Actor-Critic Methods for Offline Reinforcement Learning," 2021.

[25] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, "Bridging the gap between value and policy based reinforcement learning," Adv. Neural Inf. Process. Syst., vol. 2017-Decem, no. Nips, pp. 2776-2786, 2017.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," pp. 1-12, 2017.

[27] M. Hausknecht, P. Stone, and O. Mc, "On-Policy vs. Off-Policy Updates for Deep Reinforcement Learning," Ijcai, 2016.

[28] L. Buşoniu, R. Babuška, and B. De Schutter, "Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews, 38(2):156, 2008.," IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., vol. 38, no. 2, pp. 156-172, 2008.

[29] T. Chu, J. Wang, L. Codeca, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 3, pp. 1086-1095, 2020.

[30] OpenAI et al., "Dota 2 with Large Scale Deep Reinforcement Learning," 2019.

[31] R. R. Kumar and P. Varakantham, "On solving cooperative MARL problems with a few good experiences," arXiv, 2020.

[32] M. Tan, "Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents," Mach. Learn. Proc. 1993, pp. 330-337, 1993.

[33] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems," Knowl. Eng. Rev., vol. 27, no. 1, pp. 1-31, 2012.

[34] L. Kraemer and B. Banerjee, "Multi-agent reinforcement learning as a rehearsal for decentralized planning," Neurocomputing, vol. 190, pp. 82-94, 2016.

[35] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications," IEEE Trans. Cybern., vol. 50, no. 9, pp. 3826-3839, 2020.

[36] G. Chen, "A new framework for multi-agent reinforcement learning - centralized training and exploration with decentralized execution via policy distillation," Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS, vol. 2020-May, pp. 1801-1803, 2020.

[37] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games," 2021.

[38] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," pp. 1-20, 2018.

[39] N. Pongsakornsathien et al., "A performance-based airspace model for unmanned aircraft systems traffic management," Aerospace, vol. 7, no. 11, pp. 1-25, 2020.