A Two-Stage Reinforcement Learning Approach for Multi-UAV Collision Avoidance Under Imperfect Sensing

A 两种阶段强化学习方法在多无人机避障中的应用: 在不完美感知下的研究

Dawei Wang, Tingxiang Fan, Tao Han (R), and Jia Pan (R)
戴伟王，樊廷祥，韩涛 (R)，潘佳 (R)

Abstract-Unlike autonomous ground vehicles (AGVs), unmanned aerial vehicles (UAVs) have a higher dimensional configuration space, which makes the motion planning of multi-UAVs a challenging task. In addition, uncertainties and noises are more significant in UAV scenarios, which increases the difficulty of autonomous navigation for multi-UAV. In this letter, we proposed a two-stage reinforcement learning (RL) based multi-UAV collision avoidance approach without explicitly modeling the uncertainty and noise in the environment. Our goal is to train a policy to plan a collision-free trajectory by leveraging local noisy observations. However, the reinforcement learned collision avoidance policies usually suffer from high variance and low reproducibility, because unlike supervised learning, RL does not have a fixed training set with ground-truth labels. To address these issues, we introduced a two-stage training method for RL based collision avoidance. For the first stage, we optimize the policy using a supervised training method with a loss function that encourages the agent to follow the well-known reciprocal collision avoidance strategy. For the second stage, we use policy gradient to refine the policy. We validate our policy in a variety of simulated scenarios, and the extensive numerical simulations demonstrate that our policy can generate time-efficient and collision-free paths under imperfect sensing, and can well handle noisy local observations with unknown noise levels.

摘要-与自动地面车辆 (AGVs) 不同，无人机 (UAVs) 具有更高维度的配置空间，这使得多无人机的运动规划成为一个具有挑战性的任务。此外，在无人机场景中，不确定性和噪声更为显著，这增加了多无人机自主导航的难度。在本文中，我们提出了一种基于两阶段强化学习 (RL) 的多无人机避障方法，该方法不需要显式建模环境中的不确定性和噪声。我们的目标是利用局部噪声观测训练一种策略，以规划无碰撞轨迹。然而，强化学习的避障策略通常存在高方差和低重复性问题，因为与监督学习不同，RL 没有带有真实标签的固定训练集。为了解决这些问题，我们为基于 RL 的避障引入了一种两阶段训练方法。在第一阶段，我们使用监督训练方法优化策略，损失函数鼓励智能体遵循众所周知的互惠避障策略。在第二阶段，我们使用策略梯度来细化策略。我们在多种模拟场景中验证了我们的策略，大量数值模拟表明，我们的策略能够在不完美感知下生成高效且无碰撞的路径，并能很好地处理未知噪声水平的噪声局部观测。

Index Terms-Collision avoidance, deep learning in robotics and automation.
索引术语-避障，机器人与自动化中的深度学习。

I. INTRODUCTION

I. 引言

WITH THE widely increasing application of unmanned aerial vehicles in tracking [1], [2], disaster rescue [3] and environment exploration [4], collision avoidance algorithm for aerial robots becomes more and more important. However, the complicated and diversified workspace for aerial robots in real-world has posted a great challenge for the robustness of the multi-UAV systems since any minor error in aerial robots may cause great damage or loss for persons or other properties. Some researchers proposed solutions to multi-UAV collision avoidance systems based on centralized algorithms, which rely on a central server to communicate with each agent, and to generate global control commands according to the global observations for all robots [5]-[7]. However, the centralized multi-UAV system's reliance on communication makes it difficult to be deployed in practice, because it is usually difficult or even infeasible to maintain stable communication in large-scale and complicated scenarios due to issues such as radio interference and the radio shelter regions.

随着无人驾驶飞行器在跟踪 [1]、[2]、灾害救援 [3] 和环境探索 [4] 等领域的广泛应用，空中机器人的避障算法变得越来越重要。然而，现实世界中空中机器人复杂的多样化工作空间对多无人机系统的鲁棒性提出了巨大挑战，因为空中机器人的任何微小错误都可能对人员或其他财产造成巨大损害或损失。一些研究者提出了基于集中式算法的多无人机避障系统解决方案，这些方案依赖于中心服务器与每个代理进行通信，并根据对所有机器人的全局观察生成全局控制命令 [5]-[7]。但是，集中式多无人机系统对通信

的依赖使得其在实际部署中存在困难，因为在大型复杂场景中，由于诸如无线电干扰和无线电屏蔽区域等问题，通常很难甚至无法维持稳定的通信。

Concerning these difficulties, many methods for decentralized control are introduced for the multi-agent systems without communication [8], [9]. Many of these methods use a single sensor on-board for collision avoidance, e.g., one RGB-D camera is used in [10]. However, UAVs have six degrees of freedom and thus a single sensor cannot capture sufficient information for reliable 3D collision avoidance of a UAV. Some other methods, with ORCA3D [11] as a typical example, leverage the information of local neighbors for collision avoidance decision making. However, these methods assume that each agent can obtain a perfect estimation about the location of itself and its neighbors, as shown in Fig. 1(a), which unfortunately is difficult to achieve in real-world applications due to the imperfect sensing, as shown in Fig. 1(b). This limits many previous works in simulation, without being able to be deployed to multi-UAV systems in real world.

针对这些困难，许多无需通信的多代理系统 decentralized 控制方法被引入 [8]、[9]。这些方法中的许多使用单个机载传感器进行避障，例如，在 [10] 中使用了一个 RGB-D 摄像头。然而，无人机具有六个自由度，因此单个传感器无法捕获足够的信息来进行可靠的 3D 避障。其他一些方法，以 ORCA3D [11] 为典型代表，利用局部邻居信息进行避障决策。但是，这些方法假设每个代理能够获得关于自己和邻居位置的完美估计，如图 1(a) 所示，遗憾的是，由于不完美的感知，这在实际应用中很难实现，如图 1(b) 所示。这限制了之前许多工作在模拟中的应用，无法部署到现实世界的多无人机系统中。

In this letter, we proposed a decentralized collision avoidance policy by reinforcement learning, which leverages local neighbors' information to accomplish robust multi-UAV motion planning without the perfect sensing assumption. UAV motion planning has a higher DOF than 2D AGV planning, and our attempts to apply our state-of-the-art 2D collision avoidance approaches from our prior work [12]-[14] showed the 3D problem to be harder to solve and more expensive to train. Therefore, we propose two-stage reinforcement learning approach. Different from supervised learning, reinforcement learning (RL) does not have a fixed training set with ground truth labels, which makes RL methods suffer from high variance and low reproducibility [15]. To make our approach fast converge to the global minimum and reduce variance, we propose two-stage training method. The first stage is supervised training method with a loss function that encourages the agent to follow the well-known reciprocal collision avoidance strategy, which can optimize deep RL network parameters into a theoretically optimal zone [16]. The second stage is using traditional RL training method (Policy Gradient [17]) to maximize reward function and refine the policy. We demonstrate

在此信中，我们提出了一种基于强化学习的分布式碰撞避免策略，该策略利用局部邻居信息来完成无需完美感知假设的稳健多无人机运动规划。无人机运动规划的自由度高于 2D 自动引导车辆 (AGV) 规划，我们尝试将之前工作 [12]-[14] 中的最先进 2D 碰撞避免方法应用于 3D 问题，发现该问题更难解决且训练成本更高。因此，我们提出了两阶段强化学习策略。与监督学习不同，强化学习 (RL) 没有带有真实标签的固定训练集，这导致 RL 方法存在高方差和低可重复性问题 [15]。为了使我们的方法快速收敛到全局最小值并减少方差，我们提出了两阶段训练方法。第一阶段是监督训练方法，其损失函数鼓励智能体遵循众所周知的互惠碰撞避免策略，可以将深度强化学习网络参数优化到理论最优区域 [16]。第二阶段是使用传统的 RL 训练方法 (策略梯度 [17]) 来最大化奖励函数并优化策略。我们证明了

Manuscript received September 10, 2019; accepted January 19, 2020. Date of publication February 18, 2020; date of current version March 4, 2020. This letter was recommended for publication by Associate Editor Dr. A. Faust and Editor Prof. N. Amato upon evaluation of the reviewers' comments. This work was partially supported by the HKSAR General Research Fund under Grant HKU 11202119 and 11207818. (Corresponding author: Jia Pan.)

手稿于 2019 年 9 月 10 日收到，于 2020 年 1 月 19 日接受。发表日期为 2020 年 2 月 18 日；当前版本日期为 2020 年 3 月 4 日。本文经副编辑 Dr. A. Faust 和编辑 Prof. N. Amato 在评估审稿人意见后推荐发表。本研究得到了香港特别行政区一般研究基金的支持，资助编号为 HKU 11202119 和 11207818。(通讯作者:Jia Pan。)

Dawei Wang, Tingxiang Fan, and Jia Pan are with the Department of Computer Science, The University of Hong Kong, Hong Kong, China (e-mail: lawei@hku.hk; tingxiangfan@gmail.com; panjia1983@gmail.com).

Dawei Wang、Tingxiang Fan 和 Jia Pan 均任职于香港大学计算机科学系，中国香港 (电子邮件:lawei@hku.hk; tingxiangfan@gmail.com; panjia1983@gmail.com)。

Tao Han is with the Department of Biomedical Engineering, City University

汤汉，任职于城市大学生物医学工程系

Digital Object Identifier 10.1109/LRA.2020.2974648

数字对象标识符 10.1109/LRA.2020.2974648

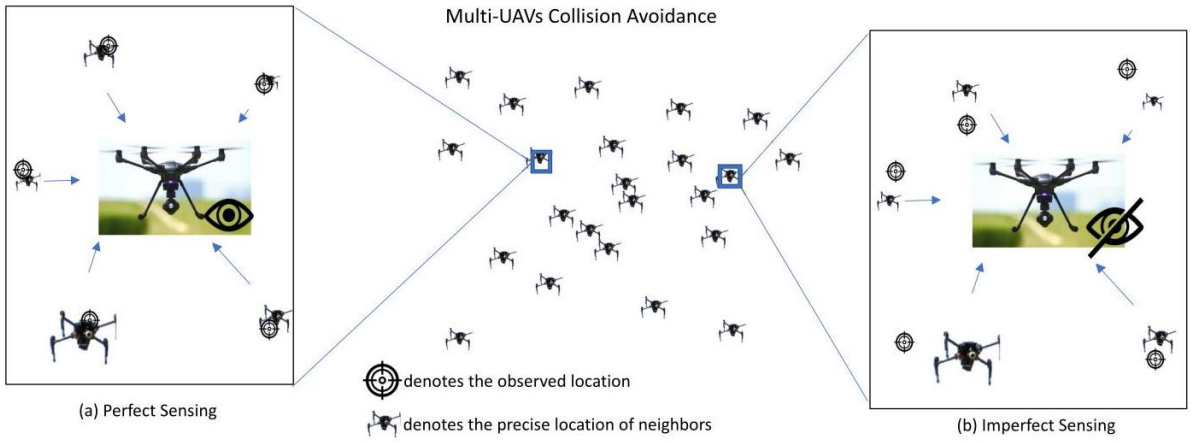

Fig. 1. Most existing collision avoidance methods assume that the environment sensing is perfect and noise-less, as shown in (a). But in the real-world environment, the sensed information is always noised and delayed (b). How to deal with such complex situation in the real-world environment is a well-known challenge for collision avoidance. In this work we propose a two-stage reinforcement learning approach to solve this challenge and demonstrate its performance in simulated scenarios where additive Gaussian noises are added to simulate imperfect sensing.

图 1. 大多数现有的避障方法假设环境感知是完美无噪声的，如图 (a) 所示。但在现实世界中，感知信息总是存在噪声和延迟 (b)。如何在现实世界环境中处理这种复杂情况是避障领域的一个知名挑战。在这项工作中，我们提出了一种两阶段强化学习方法来解决这一挑战，并在添加了高斯噪声的模拟场景中展示了其性能。

that our approach behaved robust and stable compared to the RL model without two-stage training procedure. We also validate the performance of our algorithm on unseen large-scale scenarios, which shows a good generalization of our proposed approach. The main contributions can be summarized as:

与没有两阶段训练过程的强化学习模型相比，我们的方法表现出鲁棒性和稳定性。我们还验证了我们的算法在未见过的较大规模场景上的性能，这表明了我们所提出方法具有良好的泛化能力。主要贡献可以概括为:

- We propose a decentralized reinforcement learning based policy for multi-UAV collision avoidance leveraging imperfect local observation, which provides robust performance in large-scale unseen testing scenarios with more than 200 agents in 3D workspace.
- 我们提出了一种基于分散强化学习的多无人机避障策略，利用不完美的局部观测，在大规模未见过的测试场景中提供了鲁棒性能，场景中有超过 200 个代理在三维工作空间中。
- We present a two-stage training method to reduce variance and make our approach fast converge to the global minimum.
- 我们提出了一种两阶段训练方法来减少方差，使我们的方法快速收敛到全局最小值。
- Our policy can be easily extended to handle general noise level tasks. We have trained a single policy for blind environment noise without manually adjusting parameters, which outperforms a state-of-the-art baseline.
- 我们的策略可以轻松扩展以处理一般噪声水平任务。我们训练了单个策略以应对盲环境噪声，无需手动调整参数，该策略优于现有最佳基线。

II. RELATED WORK

II. 相关工作

Long et al. [13] categorized the decentralized collision avoidance method into two types: sensor-level methods, which only feed raw on-board sensor data to the policy network, and agent-level methods, which leverage the observable states of other agents instead of raw sensor data.

Long 等人 [13] 将分散式避障方法分为两种类型: 传感器级方法，仅向策略网络提供原始机上传感器数据；代理级方法，利用其他代理的可观测状态而不是原始传感器数据。

A. Sensor-Level Navigation

A. 传感器级导航

In 2016, Bojarski et al. [18] achieved simple real-world autonomous driving by only using the raw camera data, which validated the feasibility to use deep learning for sensor-level navigation. Tai et al. [19] trained the navigation policy in the simulation via deep reinforcement learning, and then transferred the policy to the real robot. These works obtained good performance in AGVs collision avoidance problem, but they cannot be successfully applied to the collision avoidance of UAVs, which have a dimensional configuration space and also face more significant environment noises. For example, the differential wheeled robots only has two degrees of freedom to control, i.e. moving along x -axis or rotating by z -axis, and its observation can often be simplified as a few 2D points generated by a 2D LiDAR. Whereas aerial robots have six degrees of freedom of motion and thus a much larger action space, which makes decision making optimization more expensive. In addition, a single on-board sensor could only cover a very small part of the surrounding 3D environments and thus increases the observation uncertainty of the UAVs.

2016 年，Bojarski 等人 [18] 仅通过使用原始摄像头数据实现了简单的现实世界自动驾驶，这验证了使用深度学习进行传感器级别导航的可行性。Tai 等人 [19] 通过深度强化学习在模拟环境中训练导航策略，然后将策略迁移到真实机器人上。这些工作在自动引导车辆 (AGVs) 的避障问题上取得了良好的性能，但它们无法成功应用于无人机的避障，因为无人机具有维数配置空间，并且面临更显著的环境噪声。例如，差速驱动机器人只有两个自由度来控制，即沿 x 轴移动或绕 z 轴旋转，其观察结果通常可以简化为由 2D 激光雷达生成的几个 2D 点。

Some researchers presented sensor-level UAV motion planning [20] and navigation [21] algorithms. Gao et al. [22] used a stereo camera and IMU to build a teach-repeat-replan UAV system. Campos et al. [10] used a single RGB-D camera to establish an autonomous navigation framework for reaching a goal in unknown 3D cluttered environments. However, these methods could only work well in static scenarios, and UAVs' action space is also restricted for simplifying the problem. Thus, they cannot be applied to high-speed dynamic collision avoidance tasks. In addition, due to the large configuration space of a UAV, it is necessary to use more than one sensors to make a complete observation about its surrounding world. However, due to the limited battery life and carrying capability of nowadays UAVs, a successful multi-UAV collision avoidance system must survive the high observation uncertainty due to limited sensor view angles.

一些研究者提出了传感器级别的无人机运动规划 [20] 和导航 [21] 算法。Gao 等人 [22] 使用立体相机和 IMU 构建了一个教学-重复-规划无人机系统。Campos 等人 [10] 使用单个 RGB-D 相机为在未知的三维杂乱环境中到达目标建立了一个自主导航框架。然而，这些方法只能在静态场景中表现良好，并且为了简化问题，无人机的动作空间也受到限制。因此，它们不能应用于高速动态避障任务。此外，由于无人机配置空间较大，需要使用多个传感器来对其周围世界进行完整观测。然而，由于现今无人机的电池寿命和携带能力有限，一个成功的多无人机避障系统必须能够在有限的传感器视角带来的高观测不确定性下存活。

B. Agent-Level Navigation

B. 代理级别导航

Early traditional methods [23], [24] have been successfully used for crowd navigation and multi-robot collision avoidance, but they cannot robustly adapt to different scenarios due to their hand-crafted parameters. Thus, Chen et al. [25] introduced the value network of deep reinforcement learning to model the human-robot cooperative behaviors in dynamic environments. However, these approaches are all based on perfect sensing assumption and thus cannot work well in real-world applications with imperfect sensing. Recently, Tolstay et al. [26] applied Graph Neural Network (GNN) in agent-level navigation, which requires local communications during training and testing. This GNN based approach cannot be used in large-scale scenarios, because wireless communication in crowd scenarios is unreliable and any delay or unstable connection may cause collision and damage in the air. Therefore, decentralized UAVs collision avoidance, leveraging imperfect sensed agent-level information without communications between each agent, is more desirable than existing agent-level approaches.

早期的传统方法 [23]、[24] 已成功应用于人群导航和多机器人避障，但由于其手工调整的参数，无法鲁棒地适应不同场景。因此，Chen 等人 [25] 引入了深度强化学习的价值网络来模拟动态环境中的人-机器人协同行为。然而，这些方法都是基于完美感知假设，因此在感知不完美的现实世界应用中无法良好

工作。最近，Tolstay 等人 [26] 在代理级别导航中应用了图神经网络 (GNN)，这需要在训练和测试期间进行局部通信。这种基于 GNN 的方法不能用于大规模场景，因为人群场景中的无线通信是不可靠的，任何延迟或不稳定的连接都可能造成空中碰撞和损坏。因此，去中心化的无人机避障，利用不完美的感知代理级别信息且在各个代理之间无需通信，比现有的代理级别方法更受欢迎。

III. APPROACH

III. 方法

In this section, we introduce our multi-UAV reinforcement learning framework firstly. Next, we describe the network structure of the control policy for the multi-UAV system. Finally, we introduce the two-stage training method for our reinforcement learning model.

在这一部分，我们首先介绍了我们的多无人机强化学习框架。接下来，我们描述了多无人机系统的控制策略的网络结构。最后，我们介绍了我们的强化学习模型的二阶段训练方法。

A. Problem Formulation

A. 问题建模

The agent-level multi-UAV decision model can be formulated as a Partially Observable Markov Decision Process (POMDP) and can be solved using a reinforcement learning framework [12], [14]. Formally, we describe a POMDP as a 6-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O})$, where $\mathcal{S}$ is a set of states $(\mathbf{s} \in \mathcal{S}), \mathcal{A}$ is a set of actions $(\mathbf{a} \in \mathcal{A}), \mathcal{T}$ is the transition probabilities between states $\mathcal{T}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}), \mathcal{R}$ is the reward function $(\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}), \Omega$ is a set of observations $\mathbf{o} \in \Omega$, and $\mathcal{O}$ is the observation distribution given the state $\mathbf{o} \sim \mathcal{O}(\mathbf{s})$. In our formulation, the state $\mathbf{s}^t$ consists of the agents' position $\mathbf{p}$, velocity $\mathbf{v}$, and the goal $\mathbf{g}$ at t time step, i.e. $\mathbf{s}^t = \langle \mathbf{p}^t, \mathbf{v}^t, \mathbf{g} \rangle$. The action $\mathbf{a}^t$ is the steering command of a different robot in terms of linear velocities. The observation at time t is $\mathbf{o}^t$, which includes the position ${}^i o_p^t$ and velocity ${}^i o_v^t$ of the neighbors that are observable to the robot. $\mathbf{o}^t$ also includes robot' s preferred velocity ${o^t}_{\text{prefv}}$, which is the direction from its current location to its goal with the maximum speed as the magnitude. In other words, $\mathbf{o}^t = \sum_i^N \left\langle {}^i o_p^t, {}^i o_v^t \right\rangle + o_{prefv}^t$, where N is the number of the observable neighbors. The optimal policy π^* is defined according to the Bellman equation:

无人机级别的多无人机决策模型可以构建为一个部分可观测马尔可夫决策过程 (POMDP)，并可以使用强化学习框架来解决 [12]，[14]。正式地，我们将 POMDP 描述为一个六元组 $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O})$ ，其中 $\mathcal{S}$ 是状态集合 $(\mathbf{s} \in \mathcal{S}), \mathcal{A}$ 是动作集合 $(\mathbf{a} \in \mathcal{A}), \mathcal{T}$ 是状态间的转移概率 $\mathcal{T}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}), \mathcal{R}$ 是奖励函数 $(\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}), \Omega$ 是观察集合 $\mathbf{o} \in \Omega$ ，而 $\mathcal{O}$ 是在给定状态下观察的分布 $\mathbf{o} \sim \mathcal{O}(\mathbf{s})$ 。在我们的建模中，状态 $\mathbf{s}^t$ 包括代理的位置 $\mathbf{p}$ 、速度 $\mathbf{v}$ 和目标 $\mathbf{g}$ 在 t 时间步，即 $\mathbf{s}^t = \langle \mathbf{p}^t, \mathbf{v}^t, \mathbf{g} \rangle$ 。动作 $\mathbf{a}^t$ 是不同机器人以线性速度为单位的转向指令。在时间 t 的观察 $\mathbf{o}^t$ 包括机器人可观察到的邻居的位置 ${}^i o_p^t$ 和速度 ${}^i o_v^t$ 。$\mathbf{o}^t$ 还包括机器人的首选速度 ${o^t}_{\text{prefv}}$ ，这是从其当前位置到目标的指向，速度大小为最大值。换句话说，$\mathbf{o}^t = \sum_i^N \left\langle {}^i o_p^t, {}^i o_v^t \right\rangle + o_{prefv}^t$ ，其中 N 是可观察邻居的数量。最优策略 π^* 是根据贝尔曼方程定义的:

$$\pi^* = \arg\max_{\mathbf{a}^*} R\left(\mathbf{s}^t, \mathbf{a}^t\right)$$

$$+\gamma \int_{\mathbf{s}^{t+1}} \mathcal{T}\left(\mathbf{s}^{t+1} \mid \mathbf{s}^t, \mathbf{a}^t\right) V^*\left(\mathbf{s}^{t+1}\right) \mathrm{d}\mathbf{s}^{t+1}$$

$$V^*\left(\mathbf{s}^t\right) = \sum_{t'=t}^{T} \gamma^{t'-t} R\left(\mathbf{s}^{t'}, \mathbf{a}^{t'}\right), \tag{1}$$

where $R(\mathbf{s}^t, \mathbf{a}^t)$ is the reward given $\mathbf{s}^t$ and $\mathbf{a}^t$ at time t and γ is the discount factor in reinforcement learning. In this letter, the reward function is proposed as:

其中 $R(\mathbf{s}^t, \mathbf{a}^t)$ 是在时间 t 给定的奖励 $\mathbf{s}^t$ 和 $\mathbf{a}^t$ ，而 γ 是强化学习中的折扣因子。在本文中，提出的奖励函数为:

$$r^t = \begin{cases} 20 & \text{if } \left\|\mathbf{p}^t - \mathbf{g}\right\| < 0.1 \\ -20 & \text{else if collision} \\ 2.5 \cdot \left(\left\|\mathbf{p}^{t-1} - \mathbf{g}\right\| - \left\|\mathbf{p}^t - \mathbf{g}\right\|\right) & \text{otherwise.} \end{cases}$$

(2)

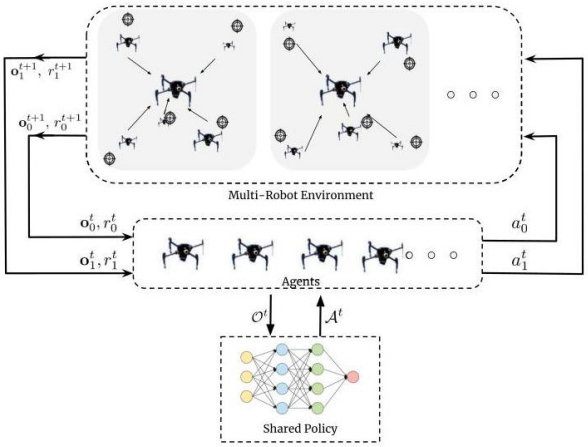

Fig. 2. Illustration of the framework of our method. Each robot obtains its neighbors' information by directly communicating with them and then leverages such information to determine control commands.

图 2. 我们方法的框架示意图。每个机器人通过直接与邻居通信获取其信息，然后利用这些信息来确定控制指令。

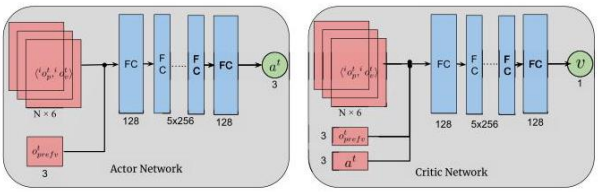

Fig. 3. The architecture of the neural network.

图 3. 神经网络的架构。

B. Multi-UAV Reinforcement Learning Framework

B. 多无人机强化学习框架

We construct a multi-robot environment with noisy sensor data and uncertainly state estimation in simulation as the data source of reinforcement learning algorithm (as shown in Fig. 2). In this environment, all agents shared the common single control policy. Overall, the data stream of the multi-UAV reinforcement learning framework is simple. The UAVs interact with the noisy environment to get all observations $\mathbf{o}_i^t$ and all reward r_i^t at time t , and these data is fed into the shared policy to compute all control commands of different robots a_i^t . Finally, these commands are executed in the multi-robot environment simultaneously, as shown in Fig. 2. All the data generated by the multi-robot environment are used for the reinforcement learning training.

我们构建了一个带有噪声传感器数据和不确定状态估计的仿真多机器人环境，作为强化学习算法的数据源 (如图 2 所示)。在这个环境中，所有代理共享单一的通用控制策略。总体而言，多无人机强化学习框架的数据流较为简单。无人机与噪声环境互动以获取所有观察值 $\mathbf{o}_i^t$ 和所有奖励 r_i^t 在时间 t ，并将这些数据输入共享策略中计算不同机器人的所有控制指令 a_i^t 。最后，这些指令在多机器人环境中同时执行，如图 2 所示。多机器人环境产生的所有数据都用于强化学习的训练。

C. Network Structure

C. 网络结构

To achieve the best performance for aerial robots, we deploy the state-of-the-art off-policy actor-critic [27] based reinforcement learning algorithm, the deep deterministic policy gradient (DDPG) [28], [29].

The architecture of our neural network consists of two networks, i.e., the actor network and the critic network, as shown in Fig. 3.

为了使无人机达到最佳性能，我们部署了最先进的基于 off-policy actor-critic [27] 的强化学习算法，即深度确定性策略梯度 (DDPG)[28]，[29]。我们的神经网络架构包括两个网络，即演员网络和评论家网络，如图 3 所示。

In the actor network, the input layer is a fully connected (FC) layer containing 256 hidden neurons, followed by the input layer, which includes five FC layers with 128 hidden neurons. Each layer uses a hyperbolic tangent (tanh) activation function. At the end, there is an output layer which also uses a fully connected structure and maps the output of the previous layer to a three-dimensional vector $\mathbf{a} = [v_x, v_y, v_z]$, which serves as the velocity control demand for UAVs.

在演员网络中，输入层是一个包含 256 个隐藏神经元的全连接 (FC) 层，其后是输入层，包含五个具有 128 个隐藏神经元的 FC 层。每一层都使用双曲正切 (tanh) 激活函数。最后，有一个输出层，它也使用全连接结构，并将前一层的输出映射到一个三维向量 $\mathbf{a} = [v_x, v_y, v_z]$ ，该向量作为无人机的速度控制需求。

In the critic network, the input layer is one FC layer containing 128 hidden neurons. After the input layer, there are five FC layers with 128 hidden neurons. The output of the critic network is the value of the value function v , generated by the output layer with a fully connected structure.

在评价网络中，输入层是一个包含 128 个隐藏神经元的 FC 层。输入层之后，有五个具有 128 个隐藏神经元的 FC 层。评价网络的输出是值函数 v 的值，由具有全连接结构的输出层生成。

D. Two-Stage Training Method

D. 两阶段训练方法

The traditional reinforcement learning algorithm is hard to train and reproduce, which has been reported, e.g., in [15]. This is because reinforcement learning does not have fixed training dataset and ground truth targets. To address such difficulty, we propose a two-stage training method for RL policy in multi-UAV collision avoidance system, where the first stage is a pre-training stage supervised by optimal reciprocal collision avoidance principle, and the second stage is a unsupervised training stage using deep deterministic policy gradient algorithm.

传统的强化学习算法难以训练和再现，这一点在文献 [15] 中已有报道。这是因为强化学习没有固定的训练数据集和真实目标。为了解决这种困难，我们为多无人机避障系统中的 RL 策略提出了一种两阶段训练方法，第一阶段是预训练阶段，由最优互斥避障原则进行监督，第二阶段是使用深度确定性策略梯度算法的无监督训练阶段。

To enable cooperative collision avoidance among multiple robots in a manner without communication, we adopt the same assumption as used in previous decentralized collision avoidance works such as ORCA [30]. In particular, we assume that all agents in the swarm will follow the same collision avoidance policy. Under this assumption, [30] proved that, if the velocity action taken by every agent falls outside the velocity obstacle zone that is constructed according to the current velocities of this agent and its neighbors, then all robots are theoretically collision-free in a given time horizon, if a perfect sensing is available.

为了在没有通信的情况下实现多机器人之间的协作避障，我们采用了与之前的去中心化避障工作 (如 ORCA [30]) 相同的假设。具体来说，我们假设群体中的所有代理都将遵循相同的避障策略。在这个假设下，[30] 证明了，如果每个代理采取的速度行动都落在根据该代理及其邻居的当前速度构建的速度障碍区域之外，那么在给定的时间范围内，如果感知是完美的，所有机器人理论上都是无碰撞的。

In an imperfect sensing scenario, we believe that the prior knowledge encoded in ORCA's velocity obstacle would also provide a strong guidance to a robust multi-agent collision avoidance policy. Our idea is to design a loss function (called "ORCA loss") based on the ORCA velocity obstacle, which is used to project the actor network's output into a reciprocal collision avoidance zone. In particular, ORCA velocity obstacle is made by a set of half-planes in the velocity space associated with each neighbor of an agent. The agent needs to optimize its navigation velocity subject to the constraint that the velocity shall fall on the negative side of all half-planes, by using linear programming. For an agent's velocity output from an actor network, its ORCA loss is computed as the distance to the closest ORCA plane of this agent's velocity obstacle. Because it has been proved that the ORCA zone is theoretically collision-free, the actor network can then converge to an optimal situation where most outputs are in the collision-free zone after several epochs training.

在不完美的感知场景中，我们相信编码在 ORCA 速度障碍中的先验知识也能为鲁棒的多人碰撞避免

策略提供强烈的指导。我们的想法是基于 ORCA 速度障碍设计一个损失函数 (称为 “ORCA 损失”)，用于将演员网络的输出投射到互斥碰撞避免区域。特别是，ORCA 速度障碍由与每个代理的邻居相关的一组半平面在速度空间中构成。代理需要优化其导航速度，使得速度落在所有半平面的负侧，通过线性规划实现。对于一个从演员网络输出的代理速度，其 ORCA 损失计算为该代理速度障碍最近 ORCA 平面之间的距离。因为已经证明 ORCA 区域在理论上是避撞的，所以演员网络可以在几个周期的训练后收敛到一个最优状态，此时大多数输出都在避撞区域内。

We implemented the ORCA loss guided pre-training in the first training stage, where the algorithm collects observations, rewards and other states information from the simulator, and computes ORCA half-planes for each agent, then stores these training data into replay buffers. The critic network will be trained as usual, while the actor network will be trained by minimizing ORCA loss, which is formally defined as:

我们在第一个训练阶段实现了基于 ORCA 损失引导的预训练，在这个阶段算法从模拟器中收集观察值、奖励和其他状态信息，并为每个代理计算 ORCA 半平面，然后将这些训练数据存储到回放缓冲区中。评判网络将像往常一样训练，而演员网络将通过最小化 ORCA 损失进行训练，该损失正式定义如下:

$$\mathcal{L}=\frac{1}{N}\sum_{i=1}^{N}\sum_{p\in P}\left(\max\left(0,-\left\|\mathbf{a}_i,p\right\|\right)\right)+\left(\mathbf{a}_i-\mathbf{v}_i^{\text{prefer}}\right), \tag{3}$$

where N is the number of agents in scenarios, P denotes the ORCA plane for each agent, $\mathbf{a}_i$ is the predicted action of agent i from actor network, $\left\|\mathbf{a}_i,p\right\|$ is the distance between $\mathbf{a}_i$ and ORCA half-plane $p,\mathbf{v}^{\text{prefer}}$ is the preferred velocity from agent's current position to its goal.

其中 N 是场景中代理的数量，P 表示每个代理的 ORCA 平面，$\mathbf{a}_i$ 是从演员网络预测的代理 i 的动作，$\left\|\mathbf{a}_i,p\right\|$ 是 $\mathbf{a}_i$ 与 ORCA 半平面 $p,\mathbf{v}^{\text{prefer}}$ 之间的距离，$p,\mathbf{v}^{\text{prefer}}$ 是从代理当前位置到目标的期望速度。

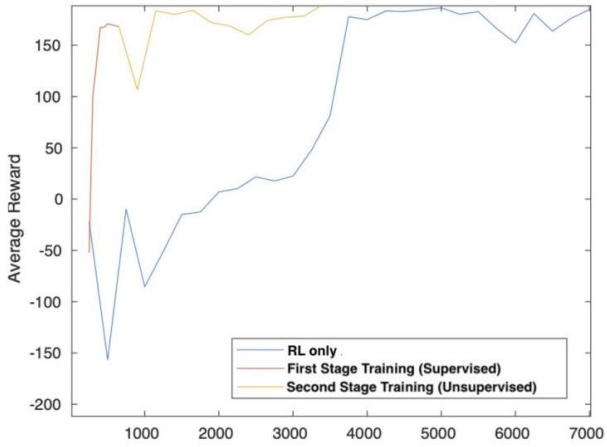

Fig. 4. Average rewards for each step during the training process.

图 4. 训练过程中每一步的平均奖励。

When the average rewards of each epoch is stable during the first training stage, the actor network and critic network both converge. Although the actor network at this moment can generate paths that are roughly collision-free, the performance of this policy is restricted by ORCA. To further increase the performance, we need to get rid of ORCA and optimize the policy by maximizing the reward function directly. Hence, we switch to the second unsupervised training stage, where we use the deep deterministic policy gradient algorithm (DDPG) [28] for training.

在第一个训练阶段，当每个周期的平均奖励稳定时，演员网络和评论家网络都会收敛。尽管此时演员网络能够生成大致无碰撞的路径，但这种策略的性能受到 ORCA 的限制。为了进一步提高性能，我们需要摆脱 ORCA，通过直接最大化奖励函数来优化策略。因此，我们转向第二个无监督训练阶段，在该阶段我们使用深度确定性策略梯度算法 (DDPG)[28] 进行训练。

Our two stage training method adapts the centralized training, decentralized inference paradigm. Specifically, all training data collected by multiple robots in simulation is used to train the control policy shared by all agents in training process. For the inference process, the control policy can be deployed to every single robot individually. This two stage training method is summarized in Algorithm 1.

我们的两阶段训练方法采用了集中训练、分散推理的模式。具体来说，多个机器人在模拟中收集的所有训练数据用于训练训练过程中所有代理共享的控制策略。对于推理过程，控制策略可以单独部署到每一个机器人上。这种两阶段训练方法在算法 1 中进行了总结。

As we can observe in Fig. 4, the average rewards increase rapidly during the first stage training period, while the traditional RL without two-stage training (RL-only) converges very slowly. Because the change of optimization goal, the average reward curve drops slightly when we switch to the second training stage, but the curve increases to a higher point very soon. Our two stage training process converges fast after about $1.5K$ steps of training, while the traditional RL takes twice the time to converge with more than $3K$ training steps.

如我们可以在图 4 中观察到，平均奖励在第一阶段训练期间迅速增加，而未经过两阶段训练的传统强化学习 (仅 RL) 收敛速度非常慢。由于优化目标的变化，当我们切换到第二阶段训练时，平均奖励曲线略有下降，但很快就会增加到更高的点。我们的两阶段训练过程在大约 $1.5K$ 步训练后迅速收敛，而传统强化学习需要超过 $3K$ 训练步才能收敛。

IV. EXPERIMENTS AND RESULTS

IV. 实验与结果

In this section, we first describe the hyper-parameters and setups in our simulator and training process. Then we illustrate the performance metrics and simulation scenarios we used for evaluation. Based on that, we compare our method with other approaches and demonstrate both the quantitative and qualitative results. Besides, we conduct additional experiments to test the

在本节中，我们首先描述了模拟器和训练过程中使用的超参数和设置。然后我们说明了用于评估的性能指标和模拟场景。在此基础上，我们比较了我们的方法与其他方法，并展示了定性和定量的结果。此外，我们还进行了额外的实验来测试我们的方法在大规模人群场景中的鲁棒性和泛化能力。

Algorithm 1: Two Stage Training of Distributed DDPG Algorithm for Multi-UAV System.

Randomly initialize the critic network $\mathbf{Q}\left(\mathbf{s}, \mathbf{a} \mid \theta^{\mathbf{Q}}\right)$ and actor network $\mu\left(\mathbf{s} \mid \theta^{\mu}\right)$ with weight θ^{Q} and θ^{μ} .

Initialize target network Q' and μ' with critic and actor network initialization weights $\theta^{Q'} \leftarrow \theta^{Q}, \theta^{\mu'} \leftarrow \theta^{\mu}$.

Initialize reply buffer R

for episode $= 1, 2, \ldots M$ do

  Receive initial observation states $s_{1,i}$

  for $t = 1, 2, \ldots T$ do

    for robot $i = 1, 2, \ldots . N$ do

      Select actions $a_{t,i}$ according to the current policy.

    end for

    Execute actions $a_{t,i}$ in the simulator, observe rewards $r_{t,i}$ and obtain new observations $s_{t+1,i}$

    Store transitions $(s_{t,i}, a_{t,i}, r_{t,i}, s_{t+1,i})$

    // sample training data from replay buffer, note that t under this line denotes the t of sampled data

    Sample a random batch of transitions (s_t, a_t, r_t, s_{t+1}) from R

    Set $y_t = r_t + \gamma Q'\left(s_{t+1}, \mu'\left(s_{t+1} \mid \theta^{\mu'}\right) \mid \theta^{Q'}\right)$

    Update critic network by minimizing the loss: $\mathcal{L} = \frac{1}{N} \sum_{t} \left(y_t - Q\big(\left(s_t, a_t \mid \theta^{Q}\right)\big)\right)^2$

    if episode $< E$ then

      // First Stage: Supervised by ORCA principle

      Update actor policy by minimizing the ORCA plane loss:

      $\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \sum_{p \in P} \left(\max\left(0, -\left\|\mathbf{a}_i, p\right\|\right)\right) + \left(\mathbf{a}_i - \mathbf{v}_i^{\text{prefer}}\right)$

```
        else
          // Second Stage: Policy gradient training
          Update the actor policy using the sampled policy
          gradient:
          ∇_{θ^μ} J ≈ (1/N) Σ_t ∇_a Q(s, a|θ^Q)|_{s=s_t, a=μ(s_t)}
          ∇_{θ^μ} μ(s | θ^μ)|_{s_t}
        end if
        if episode %τ == 0 then
          Update target networks:
          θ^{Q'} ← θ^Q; θ^{μ'} ← θ^μ
        end if
      end for
    end for
```

robustness and generalization capability of our method in large-scale crowd scenarios.
我们的方法在大规模人群场景中的鲁棒性和泛化能力。

A. Simulation Environment and Training Setup

A. 模拟环境和训练设置

To build simulation environment for multi-robot training, we implement our algorithm with ROS kinetic and Gazebo 9.0 [31]. We build our policy model based on TensorFlow [32] and train it on a PC with Intel i7-8700 k and NVIDIA GTX 1080ti. During the training process, the learning rate is fixed as $1e-3$. We set the parameters in our Algorithm 1 as $\gamma = 0.99, \tau = 400$ and $E = 3$. The simulation environment is set as a ball region whose radius is 50 m . The neighbor distance of ORCA [30] and our algorithm are both set as 10 m . Thus the agent will ignore the neighbors beyond such distance in its observation. We also limit the agent's maximum neighbor number to be 5 . In other words, only the nearest 5 neighbors information will be leveraged for navigation and collision avoidance by the agent.

为构建多机器人训练的仿真环境，我们使用 ROS kinetic 和 Gazebo 9.0 [31] 实现了我们的算法。我们的策略模型基于 TensorFlow [32] 构建，并在配备 Intel i7-8700 k 和 NVIDIA GTX 1080ti 的 PC 上进行训练。在训练过程中，学习率固定为 $1e-3$ 。我们在算法 1 中设置的参数为 $\gamma = 0.99, \tau = 400$ 和 $E = 3$ 。仿真环境设置为半径为 50 m 的球区域。ORCA [30] 和我们的算法的邻居距离都设置为 10 m 。因此，代理将忽略在其观察范围内的此距离以外的邻居。我们还限制了代理的最大邻居数量为 5。换句话说，只有最近的 5 个邻居信息将被代理用于导航和避障。

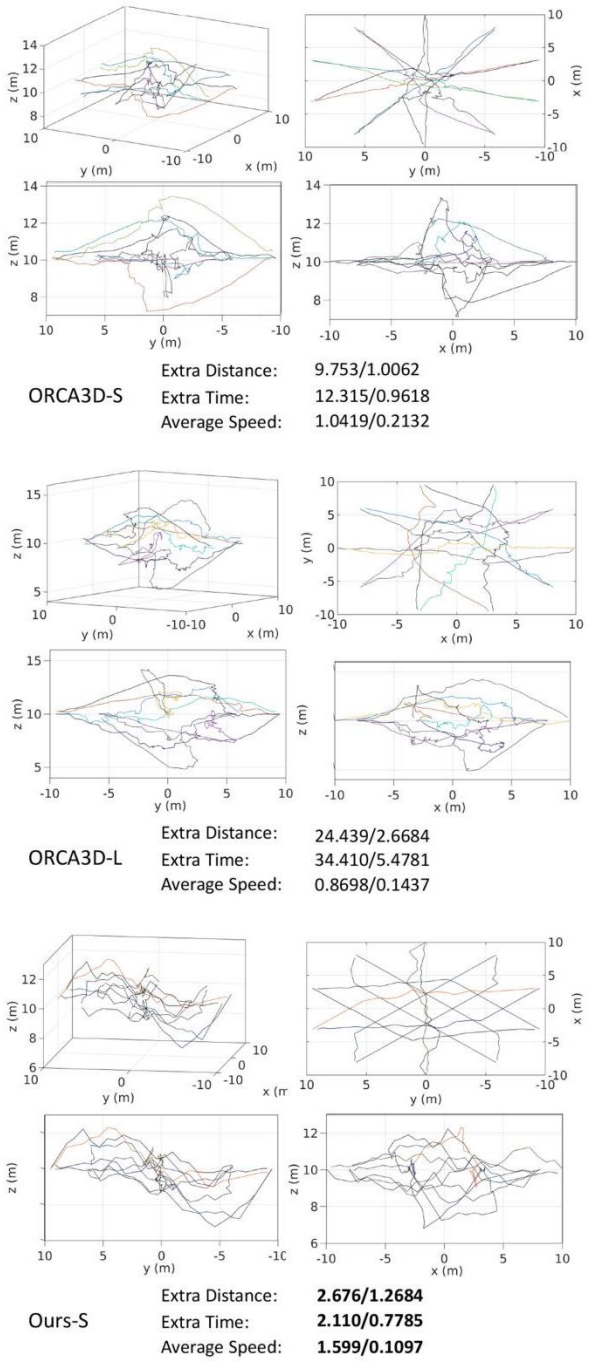

Fig. 5. Illustration of the trajectories of ORCA3D-S, ORCA3D-L and our policy under circle scenario (noise level = 0.5 , number of agents = 10) in three-view drawing and perspective drawing. The performance metrics are shown as mean/std among all test cases. Our policy produce more cooperative behavior and economical trajectory comparing to ORCA3D.

图 5. 在圆形场景下 (噪声水平 = 0.5 ，代理数量 = 10)ORCA3D-S、ORCA3D-L 和我们的策略的轨迹示意 (三视图绘制和透视图绘制)。性能指标显示为所有测试用例的平均值/标准差。与 ORCA3D 相比，我们的策略产生了更具合作性和经济性的轨迹。

In order to simulate imperfect sensing environment, we add Gaussian noise on the agent's observation, i.e., its neighbors' positions and velocities: $\mathbf{o}^{\Phi} = \mathbf{o} + \Phi, \Phi \sim \mathcal{N}\left(0, \sigma^2\right)$, where *ob* is the original observation of each agent, Φ represents the Gaussian noise follow the standard normal distribution, and $\mathbf{o}^{\Phi}$ denotes the actual noisy observation received by the agent.

为了模拟不完美的感知环境，我们在代理的观察中加入高斯噪声，即其邻居的位置和速度: $\mathbf{o}^{\Phi} = \mathbf{o} + \Phi, \Phi \sim \mathcal{N}\left(0, \sigma^2\right)$ ，其中 *ob* 是每个代理的原始观察，Φ 表示遵循标准正态分布的高斯噪声，$\mathbf{o}^{\Phi}$ 表示

代理接收到的实际带噪声的观察。

TABLE I

PERFORMANCE METRICS (SHOWN AS MEAN/STD) EVALUATED FOR DIFFERENT METHODS ON DIFFERENT SCENARIOS WITH GAUSSIAN NOISE LEVEL = 1AND NUMBER OF AGENTS = 10 . THE BEST RESULTS IN EACH CATEGORY ARE IN BOLD. SEE SEC. IV-B FOR MORE INFORMATION ABOUT EVALUATION METRICS

评估不同方法在不同场景下高斯噪声水平 = 1AND 代理数量 = 10 的性能指标 (显示为均值/标准差)，每个类别中的最佳结果用粗体表示。有关评估度量的更多信息，请参见第 IV-B 节。

$\sigma = 1$, Agent Number=10						
Scenarios	Metric	ORCA3D-S	ORCA3D-L	RL-only	Ours-S	Ours-B
Circle	Success Rate	0.96/0.894	0.985/0.051	0.9/0.106	0.985/0.082	0.98/0.185
	Extra Time	21.590/2.210	57.019/5.941	2.269/0.754	2.136/0.975	3.578/1.586
	Extra Distance	19.690/1.888	57.152/4.674	3.077/0.180	3.976/0.435	4.852/1.643
	Average Speed	1.189/0.008	0.981/0.003	1.620/0.129	1.691/0.113	1.607/0.136
Ball	Success Rate	0.98/0.044	1/0	0.90/0.091	1/0	1/0
	Extra Time	29.901/5.649	104.088/19.953	2.141/1.065	2.569/0.958	2.585/0.998
	Extra Distance	23.778/2.332	60.544/15.684	3.096/0.754	3.625/1.862	3.893/1.268
	Average Speed	1.189/0.007	0.828/0.150	1.645/0.107	1.672/0.128	1.618/0.115
Random	Success Rate	0.983/0.040	1/0	1/0	1/0	1/0
	Extra Time	15.682/3.001	150.299/7.352	2.754/0.536	1.690/0.383	2.272/1.538
	Extra Distance	16.965/3.827	70.128/20.333	1.698/0.835	1.765/0.431	1.765/0.431
	Average Speed	1.217/0.221	0.551/0.091	1.532/0.028	1.706/0.139	1.651/0.103

$\sigma = 1$, 代理数量 =10						
场景	度量	ORCA3D-S	ORCA3D-L	仅 RL	我们的方法-S	Ours-B
圆形	成功率	0.96/0.894	0.985/0.051	0.9/0.106	0.985/0.082	0.98/0.185
	额外时间	21.590/2.210	57.019/5.941	2.269/0.754	2.136/0.975	3.578/1.586
	额外距离	19.690/1.888	57.152/4.674	3.077/0.180	3.976/0.435	4.852/1.643
	平均速度	1.189/0.008	0.981/0.003	1.620/0.129	1.691/0.113	1.607/0.136
球	成功率	0.98/0.044	1/0	0.90/0.091	1/0	1/0
	额外时间	29.901/5.649	104.088/19.953	2.141/1.065	2.569/0.958	2.585/0.998
	额外距离	23.778/2.332	60.544/15.684	3.096/0.754	3.625/1.862	3.893/1.268
	平均速度	1.189/0.007	0.828/0.150	1.645/0.107	1.672/0.128	1.618/0.115
随机	成功率	0.983/0.040	1/0	1/0	1/0	1/0
	额外时间	15.682/3.001	150.299/7.352	2.754/0.536	1.690/0.383	2.272/1.538
	额外距离	16.965/3.827	70.128/20.333	1.698/0.835	1.765/0.431	1.765/0.431
	平均速度	1.217/0.221	0.551/0.091	1.532/0.028	1.706/0.139	1.651/0.103

During the experiments, we simulate two different cases of the observation noise to train our policy model. In the first case, the Gaussian noise Φ is generated based on a known parameter σ . In particular, we consider two noise levels, including $\sigma = 0.5$ and $\sigma = 1$, and train different policy models for different noise levels respectively. We refer to our models trained with known noise levels as Ours-S. In the second case, we generate the Gaussian noise Φ based on a unknown parameter σ . To achieve this, we randomly sample the noise parameter σ from the range $[0, 1]$ and generate each noisy observation independently based on different σ samples. Then we train a single policy model based on the observations containing a wide variety of unknown noise levels. We refer to such unified model trained under blind noise as Ours-B. For performance evaluation, we test all our models and other baseline approaches under the same noise level condition as in training Ours-S.

在实验过程中，我们模拟了两种不同的观测噪声情况来训练我们的策略模型。在第一种情况下，基于已知参数 Φ 生成高斯噪声 σ 。特别是，我们考虑了两种噪声水平，包括 $\sigma = 0.5$ 和 $\sigma = 1$ ，并且分别为不同的噪声水平训练了不同的策略模型。我们将基于已知噪声水平训练的模型称为 Ours-S。在第二种情况下，我们基于未知参数 σ 生成高斯噪声 Φ 。为了实现这一点，我们从范围 $[0, 1]$ 中随机采样噪声参数 σ ，并独立地基于不同的 σ 样本生成每个带噪声的观测值。然后我们基于包含各种未知噪声水平的观测值训练了一个单一的策略模型。我们将这种在盲目噪声条件下训练的统一模型称为 Ours-B。为了性能评估，我们在与训练 Ours-S 相同的噪声水平条件下测试了我们的所有模型和其他基线方法。

For ORCA policy, we found that its performance is influenced by a series of hyper-parameters. The most important one is the "Safety Space" parameter, which denotes the minimum distance of each agent keeping away from its neighbors. When we increase the "Safety Space" value in ORCA, the agent will prefer trajectories farther away from other agents and thus the probability of collision will be

lower. However, the travel distance and time cost will then increase significantly, and vice versa. Thus, one needs to balance the safety and efficiency of the algorithm when determining the "Safety Space" parameter. In our experiments, we set "Safety Space" parameter based on the noise level in testing scenarios. Specifically, we set up two ORCA policies with different "Safety Space" values, including the ORCA3D-small-safety-space (ORCA3D-S) with safety space = 0.5σ , and the ORCA3D-large-safety-space (ORCA3D-L) with safety space = 2σ .

对于 ORCA 策略，我们发现其性能受到一系列超参数的影响。其中最重要的一个是“安全空间”参数，它表示每个代理与其邻居保持的最小距离。当我们增加 ORCA 中的“安全空间”值时，代理将倾向于选择远离其他代理的轨迹，从而碰撞的概率会降低。然而，行驶距离和时间成本将会显著增加，反之亦然。因此，在确定“安全空间”参数时，需要平衡算法的安全性和效率。在我们的实验中，我们根据测试场景中的噪声水平设置“安全空间”参数。具体来说，我们设置了两个具有不同“安全空间”值的 ORCA 策略，包括安全空间为 $= 0.5\sigma$ 的 ORCA3D 小安全空间 (ORCA3D-S) 和安全空间为 $= 2\sigma$ 的 ORCA3D 大安全空间 (ORCA3D-L)。

B. Performance Metrics and Experiment Scenarios

B. 性能指标和实验场景

For performance comparison between our approach and other methods, we present the following metrics for quantitative evaluation:

为了比较我们的方法与其他方法的性能，我们为定量评估提供了以下指标:

1) Success Rate: the percentage of agents that successfully reach their own goals in the time limit without any collisions.

1) 成功率: 在时间限制内成功到达各自目标且不发生任何碰撞的代理的百分比。

2) Extra Time: the average extra travel time of agents spending on their planned trajectories compared with going straight toward their goals.

2) 额外时间: 与直接向目标前进相比，代理在其计划轨迹上花费的平均额外行驶时间。

3) Extra Distance: the average extra travel distance of agents spending on their planned trajectories compared with going straight toward their goals.

3) 额外距离: 与直接向目标前进相比，代理在其计划轨迹上花费的平均额外行驶距离。

4) Average Speed: the average speed of all agents during testing.

4) 平均速度: 测试期间所有代理的平均速度。

We employ three types of testing scenarios in our experiment:

我们在实验中使用了三种测试场景:

1) Circle Scenario: All UAVs are located uniformly on a circle at a specific altitude. Their goals will be set at the opposite side on the circle, as shown in Fig. 5.

1) 圆形场景: 所有无人机均匀地位于特定高度的圆周上。它们的目标将设置在圆的对面，如图 5 所示。

2) Random Scenario: The start and goal positions of all UAVs are initialized randomly in the simulation environment.

2) 随机场景: 所有无人机的起始和目标位置在仿真环境中随机初始化。

3) Ball scenario: All UAVs are placed randomly on the surface of a ball, and their goals are randomly placed on a concentric ball surface with a larger radius.

3) 球体场景: 所有无人机被随机放置在一个球体的表面上，它们的目标被随机放置在一个半径更大的同心球表面上。

C. Comparison on Various Scenarios

C. 各场景对比

We compare our policy with the state-of-the-art agent-level collision avoidance algorithm, ORCA policy [30] (ORCA3D-S and ORCA3D-L), and traditional RL policy without two stage training (RL-only), whose network structure is same as our proposed two-stage RL.

我们将我们的策略与最先进的个体级别碰撞避免算法，ORCA 策略 [30](ORCA3D-S 和 ORCA3D-L)，以及没有经过两阶段训练的传统强化学习策略 (RL-only) 进行了比较，后者网络结构与我们提出的两阶段 RL 相同。

Fig. 5 illustrates the trajectory of ORCA3D and our learned policy in the circle scenario. We observe that ORCA3D-L with a larger safety space behaves less economically and is more messy than ORCA3D-S

with a small safety space. It can be also seen that UAVs using our learned policy are able to move toward their goals faster than ORCA3D. Meanwhile the trajectory of our policy is uniform and symmetric, implying that our policy has learned to produce more cooperative behaviors and economical trajectories than ORCA3D.

图 5 展示了在圆形场景中 ORCA3D 和我们的学习策略的轨迹。我们观察到，具有较大安全空间的 ORCA3D-L 表现得不如具有较小安全空间的 ORCA3D-S 经济，且更加杂乱。还可以看到，使用我们学习策略的无人机能够比 ORCA3D 更快地向目标移动。同时，我们策略的轨迹均匀且对称，表明我们的策略已经学会了产生比 ORCA3D 更具合作性和经济性的行为和轨迹。

Tables I and II show the performance evaluated for different methods on different scenarios with different levels of Gaussian noises and the best results are highlighted in bold. It can be seen that the proposed policy (Ours-S) yields the best performance in most cases. For RL policy without two-stage training (RL-only), it outperforms our two-stage training RL policy in some metrics. This is because the first supervised training stage in Ours-S adopts ORCA3D's prior knowledge, In this way, our policy not only maximizes the reward function, but also tries to follow the reciprocal collision avoidance principle, which may be sub-optimal in some situations and thus may result in a longer trajectory than RL-only policy. However, our policy outperforms RL-only in success rate over all scenarios, implying that the prior knowledge from reciprocal collision avoidance principle not only helps the RL training converge fast, but also makes the policy safe and robust in unseen scenarios.

表 I 和表 II 显示了不同方法在不同场景下、不同级别的高斯噪声下的性能评估结果，最佳结果以粗体突出显示。可以看出，所提出的策略 (Ours-S) 在大多数情况下都取得了最佳性能。对于没有两阶段训练的 RL 策略 (RL-only)，它在某些指标上优于我们的两阶段训练 RL 策略。这是因为 Ours-S 的第一阶段监督训练采用了 ORCA3D 的先验知识，这样，我们的策略不仅最大化了奖励函数，还试图遵循互斥避障原则，这在某些情况下可能是次优的，因此可能导致比 RL-only 策略的轨迹更长。然而，我们的策略在所有场景下的成功率上都优于 RL-only，表明来自互斥避障原则的先验知识不仅帮助 RL 训练快速收敛，还使策略在未见过的场景中安全且健壮。

TABLE II

PERFORMANCE METRICS (SHOWN AS MEAN/STD) EVALUATED FOR DIFFERENT METHODS ON DIFFERENT SCENARIOS WITH GAUSSIAN NOISE LEVEL = 0.5AN Number of Agents = 10 . Under Noise Level = 0.5 Testing Setup, All the Methods in Every Scenarios Achieve 100% Success Rate, SO THE SUCCESS RATE METRIC IN THIS TABLE IS NOT LISTED. THE BEST RESULTS IN EACH CATEGORY ARE IN BOLD

不同方法在不同场景下、带有高斯噪声水平 = 0.5AN 代理数量 = 10 的性能指标 (显示为平均值/标准差)。在噪声水平 = 0.5 测试设置下，所有方法在每个场景下都达到了 100% 成功率，因此此表中的成功率指标未列出。每个类别中的最佳结果以粗体显示。

$\sigma = 0.5$, Agent Number =10						
Scenarios	Metric	ORCA3D-S	ORCA3D-L	RL-only	Ours-S	Ours-B
Circle	Extra Time	12.315/0.961	34.410/5.478	2.159/0.483	2.110/0.778	2.282/0.595
	Extra Distance	9.753/1.006	24.439/2.668	2.777/0.094	2.676/1.268	3.249/1.008
	Average Speed	1.087/0.0206	0.869/0.1438	1.607/0.119	1.599/0.109	1.601/0.092
Ball	Extra Time	13.833/0.779	27.797/2.718	1.668/0.357	1.582/0.437	1.983/0.517
	Extra Distance	11.226/0.869	15.301/2.811	1.726/0.301	1.896/0.827	2.581/0.882
	Average Speed	1.080/0.007	0.897/0.068	1.594/0.117	1.624/0.087	1.600/0.080
Random	Extra Time	5.257/0.951	25.439/9.122	1.034/0.257	1.567/0.465	1.472/0.583
	Extra Distance	2.692/0.984	11.192/3.026	1.173/0.582	2.108/0.843	1.965/0.738
	Average Speed	1.042/0.021	0.709/0.209	1.600/0.105	1.601/0.110	1.598/0.086

$\sigma = 0.5$ ， 代理数量 = 10						
场景	度量标准	ORCA3D-S	ORCA3D-L	仅 RL	我们的-S	我们的-B
圆形	额外时间	12.315/0.961	34.410/5.478	2.159/0.483	2.110/0.778	2.282/0.595
	额外距离	9.753/1.006	24.439/2.668	2.777/0.094	2.676/1.268	3.249/1.008
	平均速度	1.087/0.0206	0.869/0.1438	1.607/0.119	1.599/0.109	1.601/0.092
球	额外时间	13.833/0.779	27.797/2.718	1.668/0.357	1.582/0.437	1.983/0.517
	额外距离	11.226/0.869	15.301/2.811	1.726/0.301	1.896/0.827	2.581/0.882
	平均速度	1.080/0.007	0.897/0.068	1.594/0.117	1.624/0.087	1.600/0.080
随机	额外时间	5.257/0.951	25.439/9.122	1.034/0.257	1.567/0.465	1.472/0.583
	额外距离	2.692/0.984	11.192/3.026	1.173/0.582	2.108/0.843	1.965/0.738
	平均速度	1.042/0.021	0.709/0.209	1.600/0.105	1.601/0.110	1.598/0.086

TABLE III

PERFORMANCE METRICS ON THE CIRCLE SCENARIO WITH DIFFERENT TYPE OF NOISE, WITH 10 UAVS, NOISE LEVEL $\sigma = 1$. THE BEST RESULTS ARE IN BOLD

在圆形场景下，不同类型噪声、具有 10 架无人机、噪声水平 $\sigma = 1$ 的性能指标。最佳结果以粗体显示。

Method	ORCA3D-S	ORCA3D-L	Ours-S
Success Rate	0.96/0.90	1/0	0.98/0.05
Extra Distance	29.87/3.68	78.21/7.06	19.38/2.59
Extra Time	26.55/1.09	62.30/5.85	8.78/1.46
Average Speed	1.21/0.21	1.14/0.13	1.43/0.08

方法	ORCA3D-S	ORCA3D-L	我们的-S
成功率	0.96/0.90	1/0	0.98/0.05
额外距离	29.87/3.68	78.21/7.06	19.38/2.59
额外时间	26.55/1.09	62.30/5.85	8.78/1.46
平均速度	1.21/0.21	1.14/0.13	1.43/0.08

The performance of the single policy for unknown noise levels (Ours-B) is also shown in Tables I and II. As we can observe, Ours-B also achieves better performance than ORCA3D variants. In addition, unlike ORCA3D that needs to manually adjust hyper-parameters to survive in different noise levels, Ours-B can easily adapt to different unknown noise levels without parameter tuning in advance.

表 I 和 II 也展示了针对未知噪声水平 (Ours-B) 的单策略性能。正如我们所观察到的，Ours-B 相比于 ORCA3D 变体也实现了更好的性能。此外，与需要手动调整超参数以适应不同噪声水平的 ORCA3D 不同，Ours-B 可以轻松适应不同的未知噪声水平，而无需事先进行动态参数调整。

To check the policy's generalization capability to scenarios with different noises, we also evaluated the performance of different approaches in the Circle scenario under the uniform distribution noise $\frac{1}{4\sigma}[-2\sigma, 2\sigma]$ with $\sigma = 1$ and the result is shown in Table III. We can observe that the performance of all methods becomes worse than in Table I, because the noise type is very different between training and testing. However, our proposed method still outperforms baselines and survive in this new scenario with unseen noises.

为了检验策略在不同噪声场景下的泛化能力，我们还在均匀分布噪声 $\frac{1}{4\sigma}[-2\sigma, 2\sigma]$ 下，针对 $\sigma = 1$ 的圆形场景中评估了不同方法的性能，结果如表 III 所示。我们可以观察到，由于训练和测试之间的噪声类型差异很大，所有方法的性能都低于表 I。然而，我们提出的方法仍然优于基线方法，并在这个新的未见噪声场景中存活下来。

D. Large-Scale Scenario Experiments

D. 大规模场景实验

To test the performance of our proposed method on large-scale crowded scenarios, we simulate 200 UAVs in Gazebo simulator, as shown in Fig. 6. The result is shown in Table IV, which demonstrates that our learned policy can be directly applied to large-scale crowded unseen scenarios and generates better trajectory comparing to baseline methods.

为了测试我们提出的方法在大型拥挤场景中的性能，我们在 Gazebo 模拟器中模拟了 200 架无人机，如图 6 所示。结果如表 IV 所示，这表明我们学到的策略可以直接应用于大规模未见的拥挤场景，并生成比基线方法更好的轨迹。

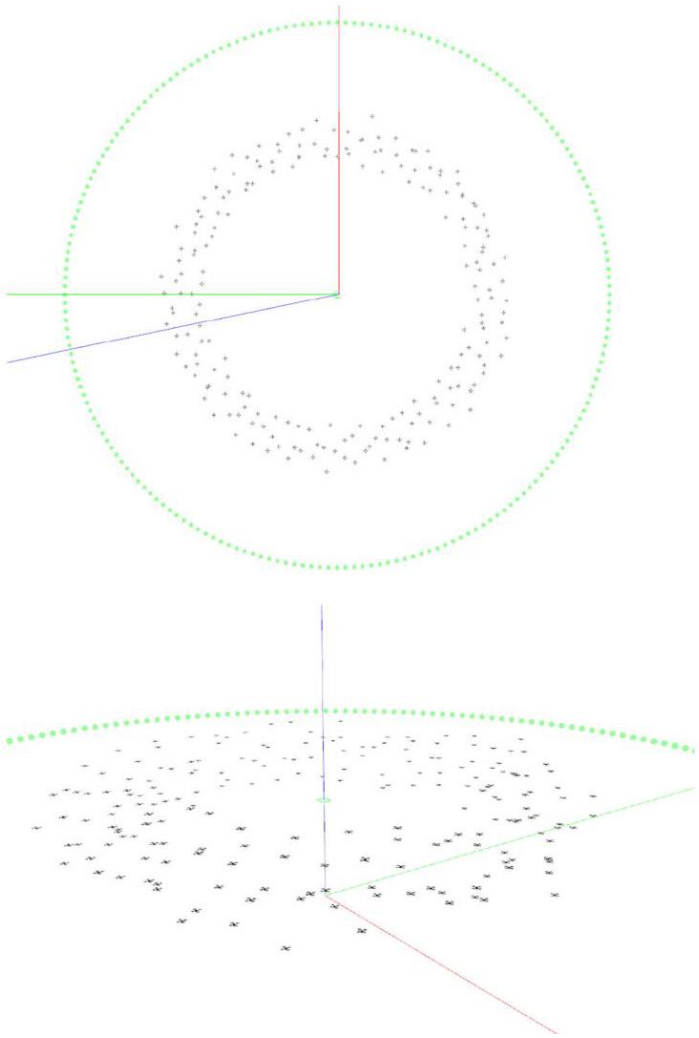

Fig. 6. Large scale scenarios with 200 UAVs in Gazebo Simulator. UAVs (black) are moving forward to their targets (green).

图 6. 在 Gazebo 模拟器中具有 200 架无人机的规模化场景。无人机 (黑色) 正向其目标 (绿色) 前进。

TABLE IV

PERFORMANCE METIRCS (SHOWN AS MEAN/STD) ON LARGE-SCALE TESTING SCENARIOS, WITH 200 UAVS, NOISE LEVEL = 1 . THE BEST RESULTS ARE IN BOLD

在大规模测试场景中，具有 200 架无人机，噪声水平 = 1 的性能指标 (显示为平均值/标准差)。最佳结果以粗体显示。

Methods	ORCA3D-S	ORCA3D-L	Ours-S
Success Rate	0.86/0.0259	0.93/0.017	0.95/0.059
Extra Distance	31.876/9.753	70.913/22.674	20.393/5.180
Extra Time	28.774/5.801	170.399/15.001	18.974/3.821
Average Speed	1.303/0.310	0.971/0.024	1.622/0.113

方法	ORCA3D-S	ORCA3D-L	Ours-S
成功率	0.86/0.0259	0.93/0.017	0.95/0.059
额外距离	31.876/9.753	70.913/22.674	20.393/5.180
额外时间	28.774/5.801	170.399/15.001	18.974/3.821
平均速度	1.303/0.310	0.971/0.024	1.622/0.113

V. CONCLUSION

V. 结论

In this letter, we presented a novel decentralized agent-level collision avoidance policy using reinforcement learning for multi-UAV system without perfect sensing assumption. We also introduced a two-stage training method to make our proposed policy more robust and converge fast. The proposed policy with two-stage training has been demonstrated several advantages over ORCA3D policy and RL policy without two-stage training in terms of success rate, trajectory length and cost of time. Benefited from strong generalization capacity of the deep neural network, our policy can generalize to large-scale unseen scenarios without fine-tuning or retraining. In contrast to benchmark methods which need to manual adjust hyper-parameters for certain observation noise level, our single policy also has the capacity to handle unknown noise level.

在此信件中，我们提出了一种新颖的去中心化代理级别的碰撞避免策略，该策略采用强化学习，适用于无需完美感知假设的多无人机系统。我们还引入了一种两阶段训练方法，以使所提出的策略更加健壮且快速收敛。与 ORCA3D 策略和未经两阶段训练的 RL 策略相比，所提出的两阶段训练策略在成功率、轨迹长度和时间成本方面具有多项优势。得益于深度神经网络的强大泛化能力，我们的策略能够推广到大规模的未见场景，而无需微调或重新训练。与需要手动调整超参数以适应特定观测噪声水平的基础方法相比，我们的单一策略也具有处理未知噪声水平的能力。

Our proposed two-stage RL policy currently only takes into account the observation at the current time and thus may have oscillation when the scenario changes abruptly due to moving obstacles. Some recent work such as [33] has demonstrated that it is important to use recurrent neural network structure to encode historical information during the navigation for reducing oscillations, and we would love to explore different recurrent network structures to solve the oscillation problem.

我们提出的两阶段 RL 策略目前仅考虑当前时间的观测值，因此在场景因移动障碍物突然变化时可能会产生振荡。近期的一些工作，如 [33]，已经证明在导航过程中使用循环神经网络结构编码历史信息以减少振荡是重要的，我们希望探索不同的循环网络结构来解决振荡问题。

REFERENCES

参考文献

[1] H. Liu, Y. Tian, F. L. Lewis, Y. Wan, and K. P. Valavanis, "Robust formation tracking control for multiple quadrotors under aggressive maneuvers," Automatica, vol. 105, pp. 179-185, 2019.

[2] H. Liu, W. Zhao, S. Hong, F. L. Lewis, and Y. Yu, "Robust backstepping-based trajectory tracking control for quadrotors with time delays," IET Control Theory Appl., vol. 13, no. 12, pp. 1945-1954, 2019.

[3] J. L. Baxter, E. Burke, J. M. Garibaldi, and M. Norman, "Multi-robot search and rescue: A potential field based approach," in Proc. Auton. Robots Agents, 2007, pp. 9-16.

[4] S. Thrun and Y. Liu, "Multi-robot slam with sparse extended information filers," in Proc. Int. Soundex Reunion Registry, 2005, pp. 254-266.

[5] J. S. Bellingham, M. Tillerson, M. Alighanbari, and J. P. How, "Cooperative path planning for multiple uavs in dynamic and uncertain environments," in Proc. 41st IEEE Conf. Decis. Control, 2002, vol. 3, pp. 2816-2822.

[6] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in Proc. IEEE Int. Conf. Robot. Autom., 2012, pp. 477-483.

[7] K. H. Movric and F. L. Lewis, "Cooperative optimal control for multi-agent systems on directed graph topologies," IEEE Trans. Autom. Control, vol. 59, no. 3, pp. 769-774, Mar. 2014.

[8] W. Liu, W. Gu, W. Sheng, X. Meng, Z. Wu, and W. Chen, "Decentralized multi-agent system-based cooperative frequency control for autonomous microgrids with communication constraints," IEEE Trans. Sustain. Energy, vol. 5, no. 2, pp. 446-456, Apr. 2014.

[9] M. Abouheaf, W. Gueaieb, and F. Lewis, "Online model-free reinforcement learning for the automatic control of a flexible wing aircraft," IET Control Theory Appl., vol. 14, no. 1, pp. 73-84, 2020.

[10] L. C.-Macías, R. A.-López, R. de la Guardia, J. I. P.-Vilchis, and D.

[10] L. C.-Macías, R. A.-López, R. de la Guardia, J. I. P.-Vilchis, 和 D.

G.-Gutiérrez, "Autonomous navigation of mavs in unknown cluttered environments,"2019, arXiv:1906.08839.

[11] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, International Symposium on Robotics Research. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. Reciprocal n-Body Collision Avoidance, pp. 3-19.

[12] P. Long, W. Liu, and J. Pan, "Deep-learned collision avoidance policy for distributed multiagent navigation," Robot. Autom. Lett., vol. 2, no. 2, pp. 656-663, 2017.

[13] P. Long, T. Fanl, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in Proc. IEEE Int. Conf. Robot. Autom., 2018, pp. 6252-6259.

[14] T. Fan, P. Long, W. Liu, and J. Pan, "Fully distributed multi-robot collision avoidance via deep reinforcement learning for safe and efficient navigation in complex scenarios," 2018, arXiv:1808.03841.

[15] A. Irpan, "Deep reinforcement learning doesn't work yet," https://www.alexirpan.com/2018/02/14/rl-hard.html, 2018.

[16] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The difficulty of training deep architectures and the effect of unsupervised pre-training," in Proc. Artif. Intell. Statist., 2009, pp. 153-160.

[17] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in Proc. Conf. Neural Inf. Process. Syst., 2000, pp. 1057-1063.

[18] M. Bojarski et al., "End to end learning for self-driving cars," 2016, arXiv:1604.07316.

[19] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in Proc. Int. Conf. Intell. Robots Syst., 2017, pp. 31-36.

[20] Y. Gui et al., "Airborne vision-based navigation method for uav accuracy landing using infrared lamps," J. Intell. Robot. Syst., vol. 72, no. 2, pp. 197- 218, 2013.

[21] C.-S. Y. C.-S. Yoo and I.-K. A. I.-K. Ahn, "Low cost GPS/INS sensor fusion system for UAV navigation," in Proc. Digit. Avionics Syst. Conf., 2003, vol. 2, pp. 8-A.

[22] F. Gao, L. Wang, B. Zhou, L. Han, J. Pan, and S. Shen, "Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments," 2019, arXiv:1907.00520.

[23] D. Helbing, I. Farkas, and T. Viscek, "Simulating dynamical features of escape panic," Nature, vol. 407, pp. 487-490, 2000.

[24] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in Proc. IEEE Int. Conf. Robot. Autom., 2008, pp. 1928-1935.

[25] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized noncommunicating multiagent collision avoidance with deep reinforcement learning," in Proc. IEEE Int. Conf. Robot. Autom., 2017, pp. 285-292.

[26] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," 2019, arXiv:1903.10527.

[27] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in Proc. Conf. Neural Inf. Process. Syst., 2000, pp. 1008-1014.

[28] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in Proc. Int. Conf. Mach. Learn., 2014, pp. 387-395.

[29] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, arXiv:1509.02971.

[30] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in Robotics research. Berlin, Germany: Springer, 2011, pp. 3-19.

[31] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in Proc. Int. Conf. Intell. Robots Syst., 2004, vol. 3, pp. 2149-2154.

[32] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, arXiv:1603.04467.

[33] L. Wang, W. Zhang, X. He, and H. Zha, "Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2018, pp. 2447-2456.