南京大学

研究生毕业论文
(申请硕士学位)

论文题目 基于深度强化学习的移动机器人导航研究

作者姓名 高可攀

学科、专业名称 计算机科学与技术

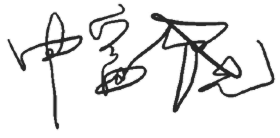
研究方向 人工智能

指导教师 申富饶

2021 年 5 月 29 日

学　　号：MG1833020

论文答辩日期：2021 年 5 月 24 日

指 导 教 师：  　　**（签字）**

Research on Mobile Robot Navigation Based on Deep Reinforcement Learning

by

Gao Kepan

Supervised by

Professor Shen Furao

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of
MASTER
in
Computer Science and Technology

Department of Computer Science and Technology
Nanjing University

May 25, 2021

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于深度强化学习的移动机器人导航研究

计算机科学与技术 专业 2018 年级硕士生姓名：高可攀

指导教师(姓名、职称)：申富饶

摘　要

导航即规划路线并控制机器人从当前位置移动至目标位置，是移动机器人的核心功能之一，随着移动机器人进入各行各业，其面对的环境更加复杂，这给机器人自主导航技术带来巨大的挑战。一方面，导航有避障和高效的基本要求，即机器人应当在运动路线上不与静态和动态障碍物发生碰撞且能尽快到达目标位置，另一方面，导航需要较强的泛化性能，因为现实场景中，传感器误差、场景变动等都会改变机器人的处境。目前虽然有许多成熟的传统导航算法，但一些算法在环境感知和理解上有所不足。深度强化学习可以赋予机器人学习和决策的能力，在机器人导航领域有巨大的应用前景。

为了实现机器人的智能避障、高效移动，使导航适用于多种场景且尽可能降低硬件计算成本，本文基于深度强化学习在移动机器人导航问题上进行深入研究，从导航的分层结构和泛化性能入手，提出了机器人强化学习导航算法和策略迁移算法，并在仿真和现实场景中都进行了验证，有力地证明了本文算法的有效性，本文的主要内容如下：

1. 本文提出了一种基于分层结构的机器人强化学习导航算法。该算法的局部规划部分基于深度确定性策略梯度算法，其端到端、无地图的特性使其不依赖建图算法和人工设计估价方案，连续动作空间的设计赋予机器人更强的机动性，轻量的网络结构使其能在低成本的硬件平台上运行。全局规划部分是强化学习与 PRM 的结合，为了优化路网建图中稀疏、耗时的问题，我们提出了基于值函数的联结方法，从而在较短时间内建立可靠的路网图。最后，为了高效训练和评估上述算法，我们构建了一套定制化仿真环境，在多个场景中的实验证实了算法的有效性和适用性。

2. 在第一个工作的基础上，本文对强化学习导航环境变换的问题进行研究，给出了跨场景导航的泛化指导并提出了基于状态空间映射和单步重构的强化学习策略迁移算法。针对强化学习导航算法泛化性能受到的挑战，在状态空间变换不大的场景间，通过增加 MDP 数量和随机噪声的方法提升导航策略的泛化性能；在变换较大的场景间，基于迁移学习进行状态空间映射的方法避免了重新训练，同时根据决策的时序性引入单步重构误差加速迁移，提升数据使用效率，相比于重新训练，迁移算法以较少的采样量得到了效果相当的策略。
3. 基于上述两种方法，本文将强化学习导航成功地应用在真实的办公场景中。结合定位、建图等算法，本文构建了一套完整的强化学习导航系统，利用 Turtlebot3-Waffle 机器人搭载该系统，在真实环境中实现了智能规划和动态避障，该系统可以作为独立模块部署在各种服务机器人中。

相关实验表明，本文提出的方法具有较强的规划、避障性能和较强的迁移能力，在真实场景中的实践也充分说明了其实用性。

关键词： 导航；机器人智能；强化学习；DDPG

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research on Mobile Robot Navigation Based on Deep Reinforcement Learning

SPECIALIZATION: Computer Science and Technology

POSTGRADUATE: Gao Kepan

MENTOR: Professor Shen Furao

ABSTRACT

Navigation means generating a route and controlling the robot to move from the current position to the destination. It is one of the core components of mobile robots. Mobile robots are entering all walks of life, bring difficulties and great challenges to autonomous navigation technology. On the one hand, navigation requires obstacle avoidance and high efficiency. The robot should not collide with static and dynamic obstacles during the movement and can reach the destination as soon as possible. On the other hand, navigation requires generalization, because sensor errors and changes of scene affect the status of the robot in the real scene. Although many traditional navigation algorithms are proposed, the algorithms lack the ability of perception and learning. Deep reinforcement learning has a strong understanding and decision-making capability. It has huge application prospects in robot navigation.

In this paper, researches navigation based on deep reinforcement learning is studied to realize intelligent obstacle avoidance and efficient movement, help the robot flexibly adapt to a variety of scenarios and reduce the computing costs. This paper focuses on hierarchical navigation structure and generalization and proposes a reinforcement learning navigation algorithm and an migration algorithm for navigation. The algorithms are verified in both virtual and real scenes, which strongly proves the effectiveness of the algorithms. The contents are as follows:

1. In this paper, a hierarchical reinforcement learning navigation algorithm is proposed. The local planner of the algorithm is based on the Deep Determin-

istic Policy Gradient algorithm. The end-to-end, no-map local planner gets rid of mapping and manual evaluations. The continuous action space provides greater mobility. The lightweight network structure enables it to deploy on a low-cost platform. The global planner is a combination of reinforcement learning and PRM. We propose a connection method based on value function to establish dense road maps rapidly. To efficiently train and evaluate the algorithms, we build a customized simulation environment. Experiments in several scenes show the effect of the algorithms.

2. The paper studies the migration of reinforcement learning navigation and gives guidance for cross-scene navigation and proposes a transfer algorithm based on state mapping and one-step reconstruction. The generalization is improved by increasing the number of MDPs and random noises between scenes with little changes in state. For scenes with large changes, the method of state mapping avoids retraining and one-step reconstruction error speeds up the transfer and improves the efficiency of data use. Compared with retraining, the transfer algorithm achieves a comparable result with a smaller sampling amount.
3. We apply the navigation system in real office scenarios. Combining localization and mapping, the paper constructs a reinforcement learning navigation system. The system deployed on a Turtlebot3-Waffle navigate intelligently and avoid dynamic obstacles in the real scene.

Experiments show that the methods proposed in this paper have a good performance in planning and obstacle avoidance. They also provide a strong migration ability for policies. Experiments in real scenes also prove the algorithms are practical.

KEYWORDS: Navigation, Robot Intelligence, Reinforcement Learning, DDPG

目　　次

插图清单

附表清单

第一章　绪论

1.1　研究背景与研究意义

随着科学技术的进步和人类社会的发展，智能机器人正逐步进入各行各业，为人类提供便利的同时也引发着产业的变革。如今，智能机器人已经遍布生产、服务、军事等各大领域，例如家庭中用于清扫的扫地机器人；商场中用于服务的导购机器人；交通物流领域中的搬运机器人；用于水下打捞、探测的深海潜水机器人；易燃易爆、有毒腐蚀、核物质等极端恶劣情况下的处理和救援机器人等。智能机器人的推广缓解了当前社会面临的诸多问题：简单劳动力不足，劳动力成本不断提升；人口结构不断变化，社会养老保障和相关服务短缺；经济水平提升，民众追求更好的生活质量等。由于上述因素，智能机器人具有巨大的发展空间和广阔的应用场景，智能机器人的研究也具有重大的意义。

移动机器人作为智能机器人的一个重要组成部分，能够进行环境感知、命令执行、导航控制、自主移动。移动机器人具备的“移动”特性使其能够胜任更多的任务。因此，如何实现移动机器人在多种场景下的自主导航成为一个重要的问题。过去几十年，众多学者对该问题进行了不同角度的各项研究，提出了不少行之有效的算法。近年来，在棋类、电子游戏上广受人们关注的强化学习在机器人导航上有巨大的应用空间。

传统的移动机器人导航包括基于已知路径的导航、基于规则的导航和基于地图的导航。基于已知路径的导航需要事先在环境中设置路标、辅助线等，导航过程中沿着路标或辅助线进行移动，这种方法需要在场景中进行复杂的标注，只适用于工厂、仓库等环境。基于规则的导航为机器人的移动设置简单的规则，比如令机器人朝着目标直线前进，如果途中遇到障碍物，则令机器人沿着障碍物顺时针或者逆时针绕行，直到返回原定的直线路径上，利用该方法实现的避障实际运动轨迹较长。基于地图的导航将在后文详细介绍，这类方法通过建立环境模型实现环境感知，在地图空间中进行搜索、采样、推演等，依赖建图算法、定位算法以及动力学模型。

(a) 物流机器人

(b) 家政机器人

(c) 扫地机器人

(d) 酒店机器人

图 1-1: 机器人导航的应用场景

相比较而言，人类往往不需要过多的外部信息，就可以前往某个目的地，例如大型商场中，在事先没有准确地图的情况下，根据一些相对位置的信息描述，人类就可以高效导航。人类这种智能的导航策略基于大脑内对环境构建的内部表征[1]，这种内部表征的构建也是人工智能算法的重要环节。深度强化学习结合了深度学习能够对高维输入进行理解表示和强化学习善于进行序贯决策的特点，在移动机器人导航中有巨大的应用前景，它能够帮助机器人像人类一样感知环境，并在与环境的交互中不断学习改进。

如图1-1所示，现今机器人的应用场景越来越多，移动机器人的自主移动能力还有很大的提升空间，结合深度强化学习提升机器人在复杂场景下的适应能力也是一大研究趋势。目前将深度强化学习用于机器人控制的各项研究不断产生新的突破，而深度强化学习与移动机器人导航的结合能够赋予机器人对环境更好的感知能力和对突发情况更好的应对能力，该方向上的研究才刚刚起步。基于深度强化学习，探索更智能的移动机器人导航，是一个有应用前景且前沿的研究。

1.2 研究现状

1.2.1 机器人智能导航研究现状

要实现移动机器人的智能化和自主化，主要考虑三个方面：一是建图，首先要记录运动场景中的特征，例如通道、障碍物的实际位置、形状、颜色等，机器

人才能感知环境；二是定位，执行任务的过程中，机器人要实时感知自身在环境中所处的位置；三是导航，也可以称为路径规划，即给定起始位置和目标位置后，机器人自主规划路径并快速安全地前往指定位置。上述的前两项任务建图和定位互相依赖，因此在实际的机器人智能运动系统中，建图和定位可以同时进行，移动机器人研究中称为同时定位与地图构建问题（Simultaneous localization and mapping，简称 SLAM）。导航建立在精确的建图和可靠的定位上，是移动机器人的关键技术。

目前广泛应用于工业场景中的导航算法大多是基于地图的，从规划的范围可以分为全局路径规划算法和局部路径规划算法两类，全局规划基于准确的整体环境信息，在较大规模的场景中粗粒度地规划出运动轨迹，而局部规划则是在环境信息未知或事先给定少量环境信息的情况下，基于传感器信息进行小规模场景下的细粒度规划。实际的导航系统中，为了实现长距离且安全的规划，往往采用全局规划与局部规划相结合的方案，全局规划给出大致的全局路径，局部规划则沿着全局路径移动并进行局部避障。

全局路径规划算法中有一类基于图的搜索算法，使用 Dijkstra、A*、D* 等算法可以在栅格地图中进行最优路径的搜索，Dijkstra 基于贪心思想，严格搜索得到最短路径，但过程中要对许多无用的节点进行计算，导致整体搜索效率低下；A* 算法采用启发式搜索，能够高效地找到估价函数下的近似最优解，但在动态场景中规划效率低；D* 算法是 A* 的优化，支持增量式搜索。全局路径规划算法中还有一类基于采样的算法，例如快速拓展随机树（Random Rapid-search Tree，简称 RRT）[2] 和概率路网图（Probabilistic Roadmaps，简称 PRM）[3]，这两种算法规划速度快，适用于对实时性要求高的大规模场景。

动态窗口法（Dynamic Window Approach，简称 DWA）[4] 是一个经典的局部规划算法，该算法在速度空间中进行采样，利用运动模型进行航迹推演，生成多条运动轨迹后进行综合评价，选择估价值最优的行动方案。人工势场法（Artifical Potential Field，简称 APF）也是局部规划算法，它令障碍物产生斥力，导航终点产生引力，计算合力驱使机器人运动，该方法收敛速度快且实时性好，但 APF 算法只在部分简单的环境中应用，在复杂的环境中会进入局部最优解，导致导航失败。

除此以外，结合兴起的人工智能算法，导航算法中产生了如蚁群算法[5]、遗

传算法[6]、神经网络法[7-8]、模糊逻辑法[9]等算法，这些算法具有一定智能性，在各种场景中都有不错的表现。

1.2.2 深度强化学习算法研究现状

强化学习的思想来源于神经科学和心理学，人类通过与自然环境的互动，不断做出适应环境的行为。强化学习是一类机器学习方法，它在智能体与环境交互的过程中不断试错，利用环境给出的奖赏或惩罚不断调整策略，引导智能体做出获取更高奖赏的行动，从而学习策略。

上世纪末，强化学习的思想就被提出，同时产生了一批基于策略的算法，如REINFORCE 等，以及基于值的算法，如 Q-Learning、SARSA 等，受限于模型的表达能力，相当长的一段时间内强化学习还只能应用在低维的任务上。但随着深度学习的发展，许多研究者提出将强化学习与深度学习结合，并形成一系列的算法，这类算法就是深度强化学习，深度强化学习利用深度网络进行值函数、策略函数的表达，取得了相当大的突破。2013 年，一项重要研究[10]将图像作为特征，用深度神经网络拟合 Q-Learning 中的状态动作值函数，即深度 Q 网络（Deep Q-Network，简称 DQN），DQN 用 Q-Learning 的值函数更新公式训练网络，同时加入经验回放机制克服数据间的相关性，最终在 Atari 游戏上取得了非常好的成绩。2015 年，DQN 又进一步加入目标网络，该技术令网络参数周期性地更新，使得训练过程更加稳定[11]。之后针对 DQN 产生了许多改进，例如优先经验回放[12]根据经验样本的优先级进行样本选取；Dueling DQN[13]改进值函数拟合方法，将 Q 值拆分为动作和状态的两部分，拟合更加准确。基于强化学习和蒙特卡洛搜索树，DeepMind 提出了围棋 AI AlphaGo[14]，在 2016 年和 2017 年战胜围棋冠军李世石和柯杰，强化学习再次受到了社会各界的广泛关注。

目前已有的强化学习方法根据是否依赖环境建模可以分为基于模型的方法和免模型的方法，也可以根据策略函数的形式分为基于值的方法和基于策略的方法等。DQN 是基于值方法的代表，输出仅限于离散动作空间，Actor-Critic 框架结合了策略梯度方法和 DQN 的 Q 网络，Actor 输出动作，Critic 评估动作。SAC（Soft Actor-Critic）[15]引入策略的最大熵，使得动作的探索能力更强，模型更稳定。深度确定性策略梯度（Deep Deterministic Policy Gradient，简称 DDPG）算法[16]将 DQN 的部分技术和 Actor-Critic 框架结合，能够输出连续动作值。A3C

（Asynchronous Advantage Actor-Critic）算法[17] 并行运行智能体，实现梯度异步更新，加快了训练速度。置信域策略优化（Trust Region Policy Optimization，简称 TRPO)[18] 和近端策略优化（Proximal Policy Optimization，简称 PPO)[19] 都通过设计约束，解决了策略梯度更新步长问题，取得了不错的表现。

深度强化学习在机器人上也有诸多应用，例如机械臂控制、机器人导航、搬运机器人控制、多机器人协同等，深度强化学习的产生促进了机器人智能算法的研究。

1.3 研究内容

为了优化导航的效果，本文分析了移动机器人导航的相关研究，围绕导航的安全性、可用性和可扩展性，展开了大量理论研究和应用实践，具体工作内容如下：

1. 本文对机器人导航问题的马尔科夫过程建模展开探讨，提出了基于分层结构的机器人强化学习导航算法。该算法的局部规划部分基于深度确定性策略梯度算法，其具有端到端、无地图的特性，从而不依赖建图算法、不依赖人工设计估价方案。另外，连续动作空间的设计也赋予机器人更强的机动性。全局规划部分是强化学习与 PRM 的结合，为了优化路网建图中稀疏、耗时的问题，我们提出了基于值函数的联结方法，从而在较短时间内建立可靠的路网图。同时，为了高效训练和评估上述算法，我们构建了一套定制化仿真环境，在多个场景中的实验证实了算法的有效性和适用性。
2. 本文对机器人导航场景变换的问题进行研究，给出了跨场景导航的泛化性能指导，提出了基于状态空间映射和单步重构的强化学习策略迁移算法。提升模型自身的泛化性能，是在状态空间变换不大的场景中应用模型的一项重要手段，对变换较大的场景无需重新训练模型，而是基于迁移学习进行状态空间映射，同时结合单步重构误差加速迁移强化学习策略，相比于重新训练，迁移算法以较少的采样量得到了效果相当的策略。
3. 本文的方法成功地应用在真实场景中。结合定位、建图等算法，本文构建了一套完整的强化学习导航系统，搭载该系统的 Turtlebot3-Waffle 机

器人，在真实环境中实现了智能规划和动态避障，该系统可以作为独立模块部署在各种服务机器人中。

1.4 论文结构

本文主要研究了复杂场景下移动机器人的二维导航，提出了基于分层结构的强化学习导航算法和机器人强化学习导航策略迁移算法，并成功地在仿真环境和现实场景中对提出的算法进行验证。全文共分为六章：第一章为绪论，主要介绍本研究的背景及意义，机器人导航和强化学习在现今的发展和本文的研究内容；第二章介绍机器人导航的相关研究工作以及强化学习的预备知识；第三章主要介绍基于分层结构的强化学习导航算法以及实验验证；第四章介绍针对机器人导航设计的强化学习策略迁移算法及其实验验证；第五章主要介绍使用本文算法构建的机器人导航系统，并将其应用在实际场景中；第六章对全文进行总结，同时展望未来工作。

第二章　相关工作

2.1　移动机器人导航

广义上来说，机器人导航包含外部感知、地图构建、机器人定位、路径规划和运动控制等系统模块，如图2-1所示。外部感知模块获取传感器输入，提取出有用的数据；环境模型模块构建导航所需要的环境信息，包括静态障碍物地图和动态障碍物地图的构建；定位模块则根据外部感知数据等指出当前机器人在地图中的位置；路径规划模块是导航的核心策略模块，它整合外部感知、地图、定位等信息，进行运动的决策；运动控制模块负责执行决策，根据运动学模型将给出的运动指令转化为对硬件的操作。

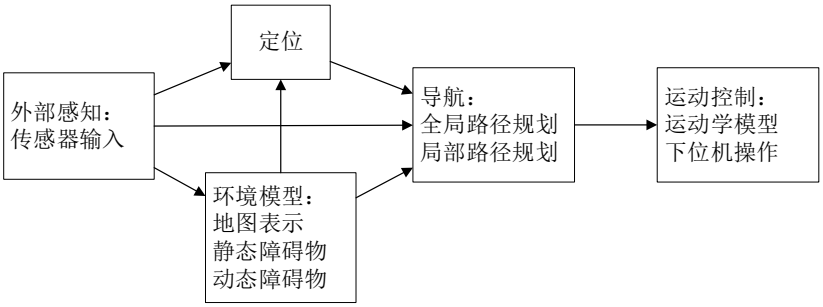


图 2-1: 机器人导航控制系统

本文研究的“机器人导航”是指在已知部分环境信息的情况下，给定起始位置和目标位置，规划路线并控制机器人运动至终点。因此本文的研究内容基于构建完备的地图和精度较高的定位系统，属于路径规划模块的范畴。对这类导航的研究主要有两个方向：全局路径规划算法和局部路径规划算法，也可以称为路径规划算法和局部避障算法。

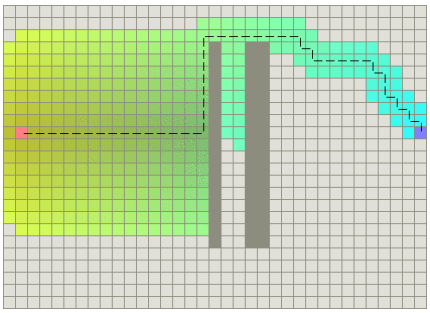
2.1.1　全局路径规划

全局规划的目标是根据静态的全局信息给出一条连接起点和终点的路径，保证路径上无障碍物。全局路径规划算法中，最典型的就是 A* 搜索算法，A*

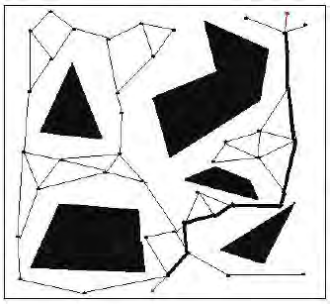
搜索算法依赖代价地图，首先将实际场景按一定尺度划分成栅格，然后根据地图中的障碍物信息衡量每个栅格受周围障碍物影响所产生的代价，构成代价地图。根据启发式函数 $f(n) = h(n) + g(n)$，A* 算法快速地在代价地图上进行规划，$h(n)$ 代表到达状态 n 花费的总代价，$g(n)$ 代表到从状态 n 到终点预估需要花费的代价，从而搜索得到连接起点栅格和终点栅格且代价总和近似最优的路径，如图2-2a所示。

A* 算法在实际使用过程中要求环境信息保持不变，如果障碍物地图发生变化，则需要重新规划。在真实环境中，机器人的运动、障碍物的移动都会使环境信息改变，降低规划效率。针对该问题提出的 D* 算法[20-21] 提高了动态环境中 A* 算法的效率，D* 算法在重规划过程中应用上一次规划访问过的中间节点信息，从而适应环境的改变，提升动态环境中的规划效率。A* 和 D* 规划算法都在完整的地图空间中进行搜索，属于确定性路径规划算法，对于联通的起点和终点，其一定能求得代价空间中的最优解，但估价函数值最小的路径在连续空间中不一定是最优的，A* 和 D* 算法给出的路径往往根据探索方向延伸，贴合障碍物，一定程度上会产生碰撞。这些算法可以用于低维环境空间的路径规划，在大规模场景、机械臂控制或者复杂动力学规划等状态空间巨大的问题中，实时性和高效性不能保障。

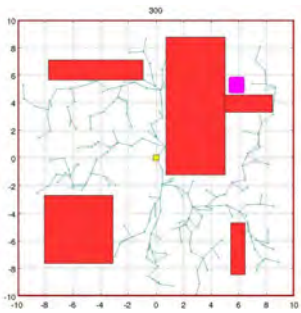
还一些基于空间采样的规划方法，这类方法降低搜索空间，从而在高维状态空间中进行规划。其中概率路网图（Probabilistic Roadmaps，简称 PRM）[3] 在地图空间按一定密度随机采样，将合法的点构成路标点集 V，然后判断这些路标点的之间的连通性，形成边集 E，从而构建路网图 $G = (V, E)$，高维空间中的规划就转化为路网图上的搜索问题，如图2-2b所示。另一种基于空间采样的规划方法是快速随机扩展树（Random Rapid-searchTree，简称 RRT）[2] 方法。该方法从起点开始生成一个树状结构，每次在空间内随机采样合法点并与周围的树节点连接，从而不断扩展树结构直到目标点，全局路径即为起点到目标点的树上路径。RRT 算法操作简单，搜索效率高，适用范围广泛。基于空间采样的算法都对原状态空间采样，构成图或树结构，然后在该结构上进行搜索，这类算法具有概率完备性，但并不保证给出最优解。相比于确定性搜索算法，它们规划效率高，因此受到广泛使用[22-23]，并有许多改进工作[24-27]。



(a) A* 规划方法



(b) 概率路网图（PRM）方法



(c) 快速随机扩展树（RRT）方法

图 2-2: 机器人全局规划算法

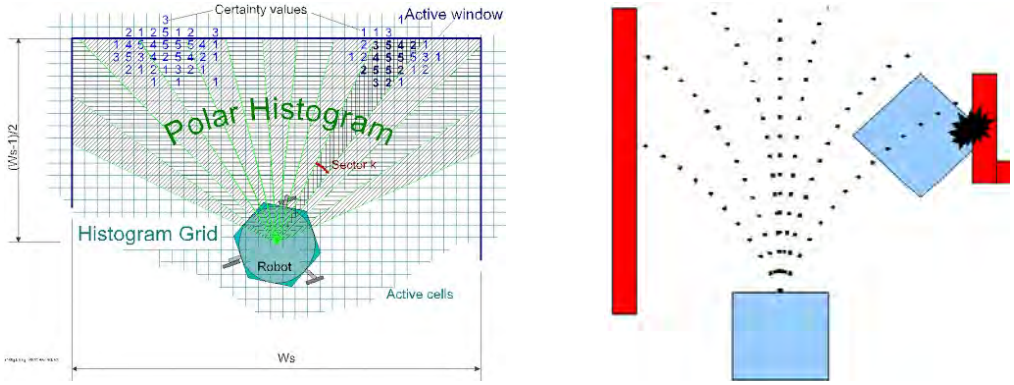
2.1.2 局部路径规划

路径规划算法的目的是基于已知的环境状态，求解出一条从起点到终点的可行路径并准确执行，实际导航过程中，由于机器人传感器误差、定位不准确和环境中出现动态障碍物等因素的影响，很难保证机器人在移动过程中，严格按照导航算法指定的路径运动，因此学者提出一类局部规划算法，利用机器人局部感知信息和地图的局部障碍物信息，在短距离内进行导航规划，从而应对环境的动态性和控制的不确定性。

常用的局部规划算法有向量直方图（Vector Field Histogram，简称 VFH）[28]、动态窗口法（Dynamic Window Approach，简称 DWA）[4] 和速度曲率法（Curvature Velocity，简称 CV）[29] 等。其中 VFH 算法最早应用于机器人局部规划，如图2-3a所示，该算法以当前机器人的位置为中心，构建局部栅格地图，并使用当前传感器的输入数据不断更新该局部栅格地图，接着将机器人的速度空间离散化为若干个扇形，选取最靠近局部规划目标的扇形区域作为运动方向，向机器人发送朝着该方向运动的速度指令。在 VFH 算法上改进得到的 VFH+ 算法[30] 考虑到机器人的运动学模型，优化速度的选择方法，从而提升机器人运动的安全性和稳定性。而 VFH*[30] 方法进行速度空间上的进一步预测，选取最优的速度指令，使用如 A* 等的搜索算法通过优化控制。

如图2-3b所示的 CV 方法和 DWA 方法的都是从速度空间优化的角度解决局部规划问题，根据不同的速度取值生成其运动空间中对应的轨迹，利用运动学模型进行轨迹生成，对每个采样速度下的理想运动轨迹进行估价，估价需要人为设计的估价函数。通常估价函数会考虑到目标位置、路径中的障碍物、机器人

位姿等信息，并对每个估价项设置系数，避障效果很大程度取决于估价函数的设计，因此这类基于估价的局部规划方法虽然操作简便，但估价函数的设计需要进行较多的对比和尝试，很多情况下估价函数不能在保证机器人绝对安全的同时保证到达目标。局部规划算法需要考虑机器人自身的运动学模型，只针对最近的传感器输入和局部的障碍物信息进行规划，因此局部规划需要较高的实时性，能够快速响应环境产生的变化。值得一提的是，局部规划在某些情况下无法导航，例如“U”形环境。



(a) 向量直方图方法 (b) DWA 方法

图 2-3: 机器人局部规划算法

2.1.3 路径规划集成

机器人导航虽然产生两个不同的研究方向，但实际应用中，往往都会将全局路径规划和局部路径规划集成在同一个导航系统中。接收导航任务后，通常导航系统会根据全局静态地图，首先使用全局路径规划（路径规划）算法输出一条从起点到终点大致可行的全局路径，然后将路径拆分为多段，将全局路径上最近的路径点作为局部导航目标，建立导航任务，交由局部路径规划（局部避障）算法完成，局部规划器根据静态和动态的环境信息，向机器人发送运动指令，最后，机器人在运行过程中，根据自身位姿的变化、场景中出现的动态障碍物以及发生改变的静态障碍物，会及时更新全局路径以及局部导航目标。全局路径规划和局部路径规划的结合使得机器人能够进行长距离导航，同时避开环境中的静态障碍物及动态障碍物。

2.2 强化学习

强化学习（Reinforcement Learning，简称 RL）是一种在时间序列上进行决策的方法，强化学习的核心——智能体（Agent）执行指定的动作（Action），同时与所处的环境（Environment）进行交互，收集当前所处环境的状态（State）及交互得到的反馈奖赏（Reward），获得的奖赏越高，说明执行的动作越好，而一系列动作执行完后获得的奖赏总和则反映了策略整体的优劣。智能体通过不断的交互、试错、学习，尽可能地最大化最终获得的总奖赏，从而优化决策能力。强化学习的简单流程如图2-4所示。

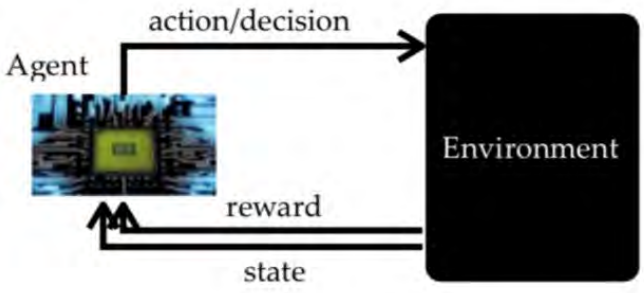


图 2-4: 强化学习的简单流程

2.2.1 马尔可夫决策过程

几乎所有强化学习的任务可以表示为马尔可夫决策过程（Markov Decidion Process, 简称 MDP）。一个马尔可夫决策过程对应四元组 $E =< S, A, P, R >$，S 表示状态的集合，A 表示动作的集合，$P : X \times A \times X \mapsto \mathbb{R}$ 表示状态之间转移的概率函数，$R : X \times A \times X \mapsto \mathbb{R}$ 表示奖赏函数。根据马尔可夫性，未来时刻的状态的概率分布仅与当前状态有关，与之前的其他状态无关，可以表示为:

$$P(s_{t+1}|s_t, s_{t-1}, ..., s_1) = P(s_{t+1}|s_t) \tag{2-1}$$

式2-1中，s_{t+1} 表示下一时刻的状态，$s_t, s_{t-1}, ..., s_1$ 表示当前时刻及先前时刻的状态。

马尔可夫决策过程的简单示意图如图2-5所示，环境的初始状态为 S_1，执行动作 a_1 后，环境的状态转移为 s_2，同时获得的奖赏为 r_1，接着不断执行上述的

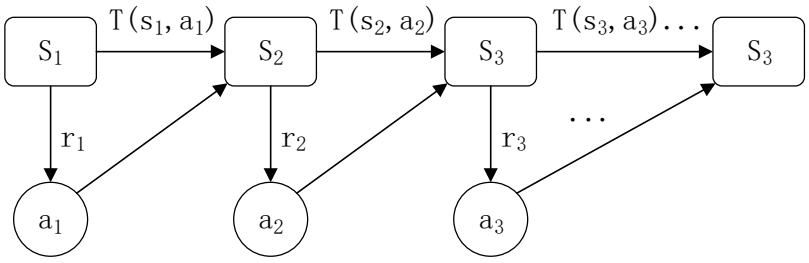


图 2-5: 马尔可夫决策过程示意

转移过程，在 t 时刻执行动作 a_t，状态由 s_t 变为 s_{t+1}。

智能体如何根据观测得到的环境状态选择动作，从而使获得的奖赏最优，就是强化学习要解决的问题。强化学习的目标是学习策略 π，该策略可以表示为一个将状态映射为动作的函数，即 $\pi : S \mapsto A$，也可以表示为在一定状态下执行动作的概率函数，即 $\pi : S \times A \mapsto \mathbb{R}$。智能体对于每一个状态都用同一个函数获取动作。

马尔可夫决策过程给出了执行单步动作的奖赏，但策略的优劣不能用单步奖赏来评判，而应该用长期执行该策略获得的累积奖赏，因此引入值函数来进行评价。值函数 $V_\pi(s)$ 描述了采用策略 π，在当前环境状态为 s 时，未来能获得的累积奖赏。累积奖赏有多种计算方式，常采用以下两种：

$$V_\pi(s) = E_\pi \left[\frac{1}{T} \sum_{t=1}^{T} r_t | s_0 = s \right] \tag{2-2}$$

$$V_\pi(s) = E_\pi \left[\sum_{t=0}^{+\infty} \gamma^t r_{t+1} | s_0 = s \right] \tag{2-3}$$

式2-2表示未来 T 步的累积奖赏，式2-3表示带有 γ 折扣的无限折扣奖赏，γ 表示折扣系数，一般 $\gamma \leq 1$，表示后续奖赏对当前的影响逐步减小。

假设在 s 状态下采取动作 a，获得的奖赏为 r_1，由于 MDP 有马尔可夫性质，式2-3可以变形为递归形式，也就是 Bellman 方程：

$$V_\pi(s) = E_\pi \left[r_1 + \gamma V_\pi(s_{t+1} | s_0 = s) \right] \tag{2-4}$$

状态值函数 $V_\pi(s)$ 表示从状态 s 出发，策略 π 能够带来的累积奖赏，如果要

进一步考虑执行动作的价值，则可以使用状态-动作值函数 $Q_\pi(s,a)$，考虑 γ 折扣奖赏，状态-动作值函数的形式为：

$$Q_\pi(s,a) = E_\pi\left[\sum_{t=0}^{+\infty}\gamma^t r_{t+1}|s_0 = s, a_0 = a\right] \tag{2-5}$$

$Q_\pi(s,a)$ 表示从状态 s 出发，执行动作 a 后，使用策略 π 能够带来的累积奖赏，同样可以变形为 Bellman 方程：

$$Q_\pi(s,a) = E_\pi\left[r_1 + \gamma Q_\pi(s_{t+1}, a_{t+1}|s_0 = s, a_0 = a)\right] \tag{2-6}$$

强化学习所要找到的最优策略就是具有最大累积奖赏的策略：

$$\pi^* = \arg\max_\pi \sum_{s\in S} V_\pi(s) \tag{2-7}$$

每个策略对应一个状态值函数和一个状态-动作值函数，最优策略对应的值函数称为最优状态值函数 $V^*(s)$ 和最优状态-动作值函数 $Q^*(s,a)$，最优值函数使对应的累积奖赏值最大：

$$V^*(s) = \max_\pi V_\pi(s) \tag{2-8}$$

$$Q^*(s,a) = \max_\pi Q_\pi(s,a) \tag{2-9}$$

根据最优状态-动作值函数，可以得到最优策略：

$$\pi^*(s) = \arg\max_{a\in A} Q^*(s,a) \tag{2-10}$$

值函数、Bellman 方程、最优策略等是马尔可夫决策过程的基本概念，大多数强化学习算法都是以此为基础进行的。主要的强化学习方法可以分为基于值函数的方法和基于策略的方法，下面对这两种方法进行简单介绍。

2.2.2 基于值函数的强化学习方法

马尔可夫决策过程中的状态值函数或状态动作值函数用于计算当前状态或决策下获得总奖赏的期望，通过值函数可以进行策略的求解和评估。式2-4是状

态值函数的 Bellman 方程，动态规划方法从该式出发，将求解每个状态的状态值定义为子问题，根据该式进行递推，迭代求解，利用上一个迭代周期的状态值函数更新当前迭代周期下的状态 s 的值。但该方法需要已知转移概率 P，因此只适用于 MDP 已知的问题，在 MDP 未知的免模型环境中，可以使用蒙特卡洛方法来估计返回值的期望，具体操作为采样获取若干轮完整的状态序列，进行经验的平均估计求取状态值，该方法每次更新值函数需要完整的一轮采样，因此学习效率低下。时序差分（TD）方法[31] 则解决了这个问题，时序差分方法使用称为自举（bootstrapping）的过程，用 $r_{t+1}+\gamma V(S_{t+1})$ 代替了 $r_{t+1}+\gamma r_{t+2}+\gamma^2 r_{t+3}+\ldots+\gamma^{T-t-1}r_T$ 并称 $r_{t+1}+\gamma V(s_{t+1})-V(s_t)$ 为 TD 误差，这样只要两个连续的状态和奖赏就能进行值函数的更新，从而提升了学习效率。

Q-Learning 算法[32] 是一个非常重要的基于值函数的强化学习方法，Q-Learning 算法时序差分求解最优状态动作值函数 Q^*，从而求解最优策略 π^*，其估算当前状态动作下 Q 值的方法如式2-11所示。Q-Learning 算法通常用于离散动作空间的问题，在离散动作下，状态动作值函数可以通过表格形式实现，算法在环境中采样 (s_t, a_t, r_t, s_{t+1}) 并对 Q 值表自身进行迭代更新，算法最后收敛时的 Q 值即为最优状态动作值函数，最优策略则根据式2-10获得，更新 Q 值的方法如式2-12所示。

$$Q_\pi(s_t, a_t) = r_t + \gamma \max_\pi Q_\pi(s_{t+1}, a_{t+1}) \tag{2-11}$$

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha[r_t + \gamma \max_{a_{t+1}} Q_\pi(s_{t+1}, a_{t+1})] \tag{2-12}$$

在环境中不断进行迭代更新，从而逼近最优状态动作值函数，Q-Learning 算法的具体过程如算法2.1所示。

2.2.3 基于策略梯度的强化学习方法

Q-Learning 算法用于处理离散动作空间的问题，连续动作空间中每个状态下可以执行无限种动作，因此对应的 $Q(s,a)$ 有无限种，很难用式2-10的方式求取最优策略。连续动作空间的问题可以采用基于策略梯度的方法，其主要思想是将策略 π 参数化表示为 π_θ，该策略直接输出动作，而不需要计算值函数，学习阶段根据状态和动作计算策略的梯度，然后将策略沿着梯度方向对动作进行优化，最终获得最优策略。

Algorithm 2.1 Q-Learning 算法

Input: π：要评估的策略，μ：采取的策略，n：训练轮数，α, γ：更新参数
Output: 最优策略 π
  for $i = 1 : n$ **do**
    初始化状态 s_t
    while 该轮未结束 **do**
      利用与策略 π 不同的策略 μ 选择动作 a_t
      执行 a_t，得到下一个状态 s_{t+1} 和奖赏 r_t
      根据式2-12更新 Q 值
      更新最优策略 $\pi(s) \leftarrow \arg\max_{a \in A} Q(s,a)$
      $s_t \leftarrow s_{t+1}$
    end while
  end for

策略梯度中的策略有随机性策略 $\pi_\theta(s,a) = P[a|s,\theta]$ 和确定性策略 $a = \pi_\theta(s)$ 两种形式，随机性策略表示当前状态为 s 下动作 a 的概率分布，确定性策略表示状态 s 唯一对应的一个动作 a。对任意马尔可夫决策过程，无论其目标函数为哪种形式的累积奖赏，根据策略梯度定理，目标函数对策略参数的梯度[33]为

$$\begin{aligned} \nabla_\theta J(\pi_\theta) &= \int_S \rho^\pi(s) \int_A Q_\pi(s,a) \nabla_\theta \pi_\theta(s,a) \, dsda \\ &= E_{s\sim\rho^\pi, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) Q^\pi(s,a)] \end{aligned} \tag{2-13}$$

根据式2-13可知，梯度 $\nabla_\theta J(\pi_\theta)$ 的计算不涉及状态概率对策略参数的梯度 $\nabla_\theta \rho^\pi(s)$，只要求出策略函数对其参数的梯度就可以进行策略的更新。在策略梯度算法中，为了进行更新，需要采样一条或多条轨迹，从而使用经验平均对策略梯度进行逼近，根据梯度上升对参数 θ 进行更新：

$$\theta \leftarrow \alpha \nabla_\theta J(\pi_\theta) \tag{2-14}$$

传统的策略梯度算法中，策略函数 $\pi_\theta(s,a)$ 可以设计为 softmax 策略函数，用 $\phi(s,a)$ 描述状态行为特征，对于离散动作空间，用式2-15表示动作的选用概率：

$$\pi_\theta(s,a) = \frac{e^{\phi(s,a)^T\theta}}{\sum_b e^{\phi(s,a)^T\theta}} \tag{2-15}$$

对于连续动作空间，动作可以从高斯分布 $\mathbb{N}(\phi(s,a)^T\theta, \sigma^2)$ 中产生。

常见的基于策略梯度的强化学习算法主要有 REINFORCE 算法[34]和算法和

Actor-Critic（简称 AC）算法[35-37] 等。Actor-Critic 算法作为一种策略梯度算法，也用到了值函数近似思想，Actor-Critic 算法包括 Actor 和 Critic 两个结构，Actor 类似于策略梯度，而 Critic 类似于 Q-Learning 等中的值函数近似。一般每经过一个回合（episode），策略梯度算法进行一次更新，所以需要得到一个回合的完整数据后才能进行学习，参数更新较慢，而 Actor-Critic 可以缓解这个问题。相比于策略梯度算法，其算法框架中设计了 Critic 结构，从而实现以步（step）为单位的更新。Critic 近似表示状态-动作值函数 $Q_\theta(s,a)$，用于近似评价在状态 s 下执行动作 a 的优劣，Critic 为 Actor 提供梯度进行更新。Actor-Critic 算法结合了值函数和策略梯度的优势，其名称可以理解为 “行动者-评论家”，行动者基于评论者的评估优化并采取行动，评论家 Critic 基于历史行动的反馈近似评估当前的行动，算法结构如图2-6所示。

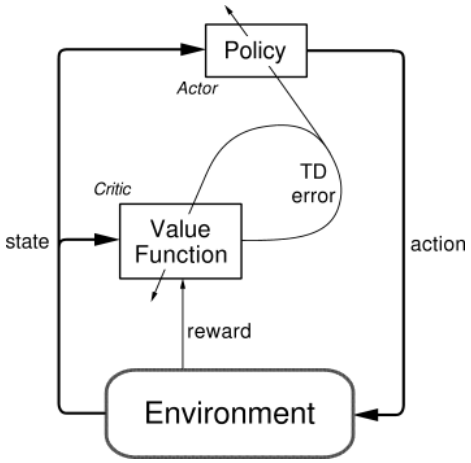


图 2-6: Actor-Critic 算法示意图

在 AC 算法中，Actor 用于输出动作，是策略函数的近似：$\pi_\theta(s,a) = P(a|s,\theta) \approx \pi(a|s)$，其中 θ 为策略的参数；Critic 用于对动作进行评估，评估对象是多样的，常见可以基于以下值进行评估：

1. 基于状态价值函数：$V(s)$
2. 基于状态-动作价值函数：$Q(s,a)$
3. 基于 TD 误差：$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ 或者 $\delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$
4. 基于优势函数：$A(s,a) = Q(s,a) - V(s)$

如果基于 TD 误差 $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$，则 Critic 对状态价值函数进行近似：$V_\omega(s,a) \approx V(s,a)$，$\omega$ 为 Critic 的参数。策略 π_θ 和 $V_\omega(s,a)$ 的参数单步更

新公式为:

$$\begin{aligned}\theta_{t+1} &= \theta_t + \alpha\delta_t \bigtriangledown_\theta log\pi_\theta(s_t, a_t) \\ &= \theta_t + \alpha\delta_t \frac{\bigtriangledown_\theta \pi_\theta(s_t, a_t)}{\pi_\theta(s_t, a_t)}\end{aligned} \tag{2-16}$$

$$\omega_{t+1} = \omega_t + \alpha\delta_t \bigtriangledown_\omega V_\omega(s_t) \tag{2-17}$$

其中 α 为学习率，θ_{t+1} 为 Actor 更新后的参数，ω_{t+1} 为 Critic 更新后的参数，$V_\omega(s_t)$ 和 $V_\omega(s_{t+1})$ 分别为将当前时刻状态 s_t 和下一时刻状态 s_{t+1} 输入 Critic 得到的状态价值。

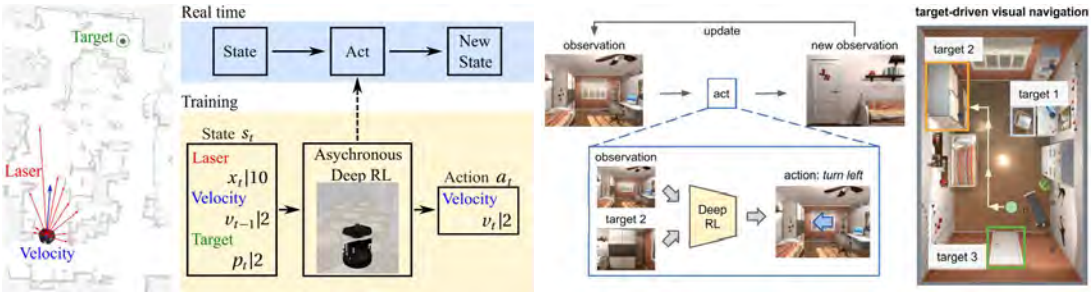
相较于策略梯度算法，Actor-Critic 算法更加高效，每与环境交互一次都可以更新参数。面对高维度的状态空间时，Actor 和 Critic 可以采用神经网络的结构，每次迭代更新时，先通过 Critic 计算 TD 误差 δt，接着分别计算 Actor 和 Critic 的梯度并更新网络的参数。Actor-Critic 算法仍然存在一定的不足，最大的缺点是不易收敛，但 Actor-Critic 框架是许多强化学习算法的基础。

2.3 基于强化学习的机器人控制

随着强化学习的理论发展，强化学习在许多领域都取得了令人瞩目的进展。近几年来，除了电子游戏、棋类，推荐系统[38]、自然语言处理[39] 等领域中都有强化学习的应用。机器人的许多控制任务都是在进行连续决策，因此机器人智能领域也有许多强化学习的实践。在机器人控制方面，OpenAI 曾发布多个环境，利用强化学习训练机器人进行操作[40]；文献[41] 利用 soft Q-learning（SQL）[42] 进行现实世界的机器人操作，SQL 具有多模态探索策略和组合创建策略两个重要特征，通过策略组合比从头开始训练更加有效。文献[43] 提出 RSI（Reference State Initialization）和 ET（Early Termination）的方法来学习多种技能，并通过多技能融合使动作更加贴近参考动作，实现了运动、杂技和武术等控制。文献[44] 针对真实足式机器人训练复杂、昂贵的问题，在仿真环境构建物理世界的动力学模型并训练神经网络策略，然后将收敛后的策略应用到 ANYmals 机器人中，实现了快速、自动化、低成本的数据生成和准确、高效的运动能力，能够实现高速奔跑，具有自主恢复能力。文献[45] 使用元学习进行机器人控制，提出机器人能够在较少的试错下就能学习策略的 micro-data RL（MDRL），在机器人强化学习中

融合先验，提高学习效率、数据利用率，从而使机器人快速适应新的环境。在机器人感知方面，文献[46]通过触觉模型预测控制（tactile model-predictive control）将强化学习用于触觉传感器，实现精细的控制。文献[47]将光流预测结合进强化学习，以处理动态的物体或者任务目标。文献[48]结合触觉和视觉实现更好的控制，使用自监督学习感知输入的紧凑、多模态表征，提升数据利用效率，并使学得的策略具有鲁棒性，可以泛化至不同的条件下。

在机器人导航方面，文献[49]从深度相机的原始输入中提取信息，采用GAIL[50]实现了动态环境中符合社会礼仪要求的避障和导航。文献[51]采用DQN算法，将原始RGB-D图像作为输入，在仿真环境中训练了更加复杂的探索策略，控制机器人保持避障的同时持续运动。文献[52]基于激光雷达输入进行强化学习导航，配合机器人的运动状态信息，利用ADDPG（Asynchronous DDPG）算法训练机器人在小规模场景中实现端到端的导航，如图2-7a所示。文献[53]利用视觉感知进行强化学习导航，采用A3C算法，基于ImageNet上预训练得到的ResNet网络和全连接层构建Actor-Critic网络，将当前相机输入的RGB图像和目标位置的图像作为输入，输出三维环境中的动作，提升算法对不同任务目标的通用性，同时采用AI2-THOR框架[54]以提升数据效率，从而实现了如图2-7b所示的室内小规模场景下，对不同目标的视觉导航。



(a) 基于激光雷达的导航 (b) 基于视觉的导航

图 2-7: 机器人导航的应用场景

2.4 本章小结

本章主要介绍了机器人导航的相关方法、强化学习的理论及其在机器人上的应用，是第三、第四章提出的相关算法的重要理论基础。对导航算法按照研究角度从全局路径规划和局部路径规划进行介绍，同时介绍了将两者结合的导航

框架。对强化学习理论论述了马尔科夫决策过程以及一些经典的基于值函数和基于策略的强化学习算法。本文第三、第四章中使用了基于策略梯度的强化学习方法，因此对基于策略的强化学习算法以及基于策略梯度的 Actor-Critic 算法进行了深入探讨。

第三章　机器人分层强化学习导航研究

本章提出了一种基于分层结构和强化学习的机器人导航算法，旨在使用深度强化学习，提供更高效且更安全的导航。本章的前半部分对机器人导航局部规划进行马尔可夫决策过程的建模，接着介绍端到端、无地图的局部强化学习导航具体设计。后半部分介绍结合 PRM 的全局强化学习-PRM 规划。最后展示导航算法的性能并进行实验分析。

3.1　局部强化学习规划

本文所要解决的机器人导航问题，是控制机器人快速从给定的起点到达给定的终点，并在移动过程中避免与障碍物碰撞的问题。为了实现长距离的导航规划，首先把整个导航问题划分为局部规划和全局规划，局部规划实现传感器视野（Field of View，简称 FOV）内的局部避障，全局规划负责找到长距离路径并将其拆分为多个局部规划问题。

除了满足安全、快速的基本要求，局部强化学习规划还具有端到端、无地图的特点。“端到端”指局部规划的输入为不需要额外处理的环境状态，输出为可以直接发送给硬件的控制指令，端到端使得该局部规划算法具有独立性，在简单场景中，即使没有全局规划算法协同，也可以独立运行，实现近距离内导航；端到端的结构也避免了复杂的运动学建模，使算法更具泛化性。“无地图”指强化学习 agent 在接收环境状态和奖赏信息并用它们进行学习和决策的过程中，没有任何地图信息，因此无法获取世界坐标系和静态障碍物分布。传统局部规划算法极大依赖地图，无论是建立全局静态障碍物地图还是局部动态障碍物地图，都需要相当大的计算量，无地图可以略去传统算法必不可少的建图过程。

3.1.1　状态空间设计

机器人导航的状态空间主要由两部分构成，分别为外部传感器的输入和机器人当前的运动状态。

为了提升导航性能，降低成本，本文使用的传感器为激光雷达（Light Detection and Ranging，简称 LiDAR）。激光雷达是当下机器人智能、自动驾驶等领域中使用最广泛的传感器设备之一，它的原理是持续不断地发射光束并接收反射，通过计算发射时间和接收时间的时间差，可以得到与障碍物之间的距离。单线激光雷达可以扫描生成二维平面内的点云信息，多线激光雷达能够生成三维空间内的点云信息，本文针对平面上移动的机器人进行导航，因此使用单线激光雷达，它的扫描角度为 360°，即整个平面。通过数据处理获得包含 360 维数据的向量，避障中不需要这么高维的数据，因此进行分区裁剪简化输入。

激光雷达的扫描如图3-1所示。从 360 个角度中等间距选取其中 n_L 个角度，取这些角度的激光雷达数据作为输入，经过处理后，构成一个 n_L 维向量。取 $n_L = 60$ 时，以机器人当前朝向为 0° 方向，逆时针每隔 6° 读取障碍物的距离，最后构成 60 维向量 $\mathbf{L}$，数据的最小值为激光雷达的最小测距，最大值为激光雷达的最大测距，如果某个方向上没有障碍物，则用一个比最大测距稍大的固定值表示。

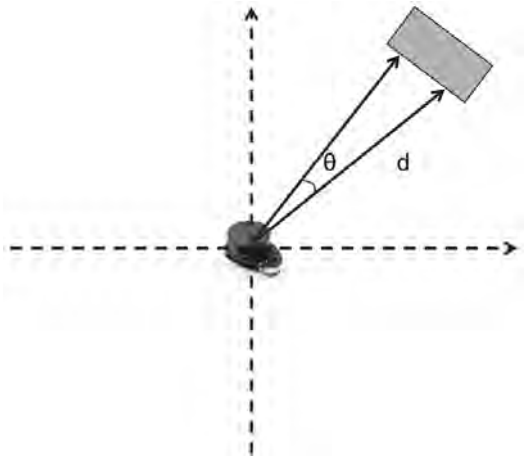


图 3-1: 激光雷达扫描示意

机器人的运动状态包括位姿和速度。在三维的机器人动力学中，常用向量 $P = [p_x, p_y, p_z]^T$ 描述机器人在空间中的位置，用 3×3 的矩阵描述机器人的姿态，再将两者结合构成 4×4 的变换矩阵，从而进行三维位姿的齐次表示。但在本节描述的导航任务中，机器人只在二维平面内运动，且规定强化学习导航过程中不能获取地图信息，也就不存在任何世界坐标系来描述起点和终点的位置。因此，以机器人当前位置为极点，当前的朝向为极轴建立极坐标系，将终点在该坐标系中的位置 (ρ, θ) 表示为机器人的位姿。机器人的速度用当前的线速度 v_l

和角速度 v_a 表示。

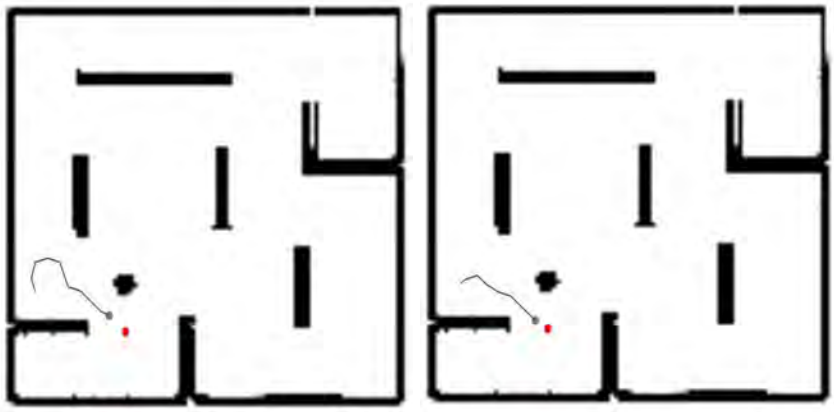
导航任务的最终状态 $\mathbf{S} = [\mathbf{L}, \rho, \theta, v_l, v_a]^T$，取 6° 为激光间隔时，状态空间是 64 维向量。

3.1.2 动作空间设计

导航任务的动作空间 $A = [v_l', v_a']$，表示要求达到的目标线速度 v_l' 和目标角速度 v_a'。$v_l' \in [-V_{max\_l}, V_{max\_l}]$，$V_{max\_l}$ 表示最大线速度，当 v_l' 为正数表示机器人向前移动，为负数表示机器人向后移动；$v_a' \in [-V_{max\_a}, V_{max\_a}]$，$V_{max\_a}$ 表示最大角速度，当 v_a' 为正数表示机器人逆时针转向，为负数表示机器人顺时针转向。

在实际运动过程中，机器人往往不能严格按照目标速度运行，因为加速或减速达到目标速度需要一定时间，这段时间不能忽略不计。值得一提的是，受机器人机械损耗或地面摩擦等因素影响，加速度可能不会是一个定值。

许多强化学习研究会对连续动作空间进行离散化以简化问题，但本文中 v_l' 和 v_a' 仍取连续值。机器人的最大线速度 V_{max_l} 和最大角速度 V_{max_a} 由硬件决定，往往会构成范围较大的动作空间，在此空间上进行离散化采样意味着仅选取若干个固定的运动动作，例如对于最大线速度 0.2m/s，最大角速度 1.0rad/s 的移动机器人来说，可以选取以下的动作：前进（线速度 0.2m/s，角速度 0rad/s）、后退（线速度-0.2m/s，角速度 0rad/s）、左转（线速度 0m/s，角速度 1.0rad/s）、右转（线速度 0m/s，角速度-1.0rad/s）等。虽然已有许多基于这类固定动作集合的研究，但固定动作集合存在若干问题：固定动作的选取是人为规定的，选取的好坏决定导航的性能；固定动作无法发挥移动机器人的机动性，例如面对较窄的弯道，仅依靠若干个固定动作可能无法通行；移动机器人具有惯性，严格按照固定动作行进前需要完全减速至静止，这样低效、耗能且易产生误差。如图3-2所示，在不同动作空间下进行同一导航任务，离散动作空间下的导航路径更曲折复杂，机动性差。



(a) 离散动作空间下的路径 (b) 连续动作空间下的路径

图 3-2: 同一导航任务在离散动作空间和连续动作空间下的路径对比

3.1.3 奖赏函数设计

一个导航任务有成功和失败两种结果，如果简单地在导航成功时给与 1 的奖赏，导航失败时给予-1 的奖赏，其他时刻奖赏为 0，就会遇到稀疏奖赏（Sparse Reward）的难题。此时探索过程中大多数情况下奖赏为 0，且训练的前期策略效果与随机策略相当，基本不可能收到正奖赏，会导致学习算法的收敛非常慢，甚至无法学习。迄今为止已有许多针对强化学习稀疏奖赏问题的研究，其中一类研究是采用分层的结构（Hierarchical Reinforcement Learning，简称 HRL），HRL 分而治之，将较复杂的问题划分为多个阶段，每当完成一个阶段的子问题后给予较高的奖励，这样就能引导 agent 逐步学会处理原问题的策略。而应对稀疏奖励，另一种简单的思路是人为设计奖赏函数，将原本的稀疏奖赏转化为稠密奖赏。

人为设计奖赏是许多强化学习任务重要的一环，奖赏函数填充了探索中值为 0 的奖赏，同时也一定程度上提供了对环境的先验信息，辅助学习算法快速收敛。在导航问题中，结合人类对导航问题的直觉，基于尽可能简化的原则，采用如下奖赏函数：

$$reward(P_{now}, P_{goal}) = \begin{cases} R_s & success \\ R_m * ||P_{now} - P_{goal}||_2 & move \\ R_f & fail \end{cases} \tag{3-1}$$

式3-1中 P_{now} 表示机器人当前位置，P_{goal} 表示终点位置，$||P_{now} - P_{goal}||_2$ 表

示当前位置和终点的欧氏距离，R_s 表示机器人到达终点后的奖赏，为正值，R_m 表示机器人每次移动后的时间惩罚参数，为负值，R_f 表示机器人导航失败后的惩罚，为负值。导航失败意味着规定时间内没有到达终点，或与障碍物发生了碰撞。R_s 和 R_f 控制机器人能否完成任务，移动惩罚项中的 $||P_{now} - P_{goal}||_2$ 要求机器人尽可能靠近目标，R_m 引导机器人快速到达终点。

奖赏函数中参数的取值对最终学得的策略有很大影响，R_s、R_m、R_f 的具体取值将在实验中说明。

3.1.4 深度确定性策略梯度算法

强化学习是一种对样本量要求很大的机器学习算法，但无论是在仿真环境还是在现实场景中，机器人行为的采样代价都很高。仿真系统中，由于机器人组件众多、运动学模型复杂，仿真计算量大，而现实中采样频率则更低。因此本章介绍的局部规划采用训练速度更快的深度确定性策略梯度算法，本节将介绍以 Actor-Critic 算法框架为基础的确定性策略梯度算法及其改进得到的深度确定性策略梯度算法。

3.1.4.1 DPG 算法

确定性策略梯度算法（Deterministic Policy Gradient Algorithms，简称 DPG）基于 Actor-Critic 框架，同时针对 Actor-Critic 算法难以收敛的缺点进行了改进。确定性策略是相对于随机性策略而言的，随机性策略中的策略函数 $\pi_\theta(s,a)$ 表示在状态 s 下采用动作 a 的概率，同时保证 $\sum_{a\in A}\pi_\theta(s,a)=1$，每次行动通过对随机策略 π_θ 进行采样得到动作，并以最大的累积奖赏为目标进行策略梯度更新。随机策略可以用于低维离散动作集合的强化学习任务中，但面对连续动作空间时，往往需要对连续空间进行离散化采样，用某个固定的动作值代表一个区间的动作值，这样的离散化操作丢失了大量动作取值。而面对高维离散动作集合时，随机性策略需要计算每个动作采用的概率并进行采样，那么计算量会非常大，产生维度灾难。

节3.1.2介绍了机器人导航的动作建模，动作空间 A 为二维连续动作空间，不选取固定动作，随机性策略梯度算法不好应对，因此，引入确定性策略梯度算法。DPG 算法的核心是采用确定性策略 $\pi_\theta(s)=a$，随机性策略输出一组动作的

采用概率，是一个状态动作的概率分布函数，而确定性策略则输出具体的动作值，是一个状态到动作的映射函数。

将策略梯度推广到确定性策略，其梯度形式为

$$\begin{aligned}\nabla_\theta J(\pi_\theta) &= \int_S \rho^\pi(s)\ \nabla_\theta\ Q^\pi(s,\pi_\theta(s))\,ds \\ &= \int_S \rho^\pi(s)\ \nabla_\theta\ \pi_\theta(s)\ \nabla_a\ Q^\pi(s,a)|_{a=\pi_\theta(s)}\,ds \\ &= E_{s\sim\rho_\pi}[\nabla_\theta\pi_\theta(s)\ \nabla_a\ Q^\pi(s,a)|_{a=\pi_\theta(s)}]\end{aligned} \tag{3-2}$$

式3-2中 $\rho^\pi(s)=\int_S\sum_{t=1}^{\infty}p_1(s)p(s\to s',t,\pi)\,ds$ 表示状态的分布，其中 $p_1(s)$ 表示 s 作为初始状态的概率，$p(s\to s',t,\pi)$ 表示采用策略 π 在 t 时刻下状态由 s 到 s' 的转移概率。相较于随机性策略梯度，确定性策略梯度在动作空间 π_θ 上不用考虑 $a=\pi_\theta(s)$ 以外的其他动作，同时在状态动作函数上对动作 a 进行了求导。

DPG 算法可以采用 Actor-Critic 的结构，构建 Critic 对状态动作函数近似，Critic 的作用和参数更新方法与随机策略梯度的 Actor-Critic 算法完全一致，通过 Critic 给出近似的评估值，接着在近似梯度方向上进行参数更新，式3-2中的状态动作值部分改用近似结果 Q_ω,。基于 TD 误差 $\delta_t=r_t+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t)$，DPG 算法的参数更新过程如下：

$$\begin{aligned}\theta_{t+1} &= \theta_t+\alpha\ \nabla_\theta\ \pi_\theta(s_t)\ \nabla_a\ Q_\omega(s_t,a_t)|_{a=\pi_\theta(s)} \\ \omega_{t+1} &= \omega_t+\alpha\delta_t\ \nabla_\omega\ Q_\omega(s_t,a_t)\end{aligned} \tag{3-3}$$

这种形式的 DPG 算法中收集数据的行为策略和被改进的目标策略是同一个，即 $a_t=\pi_\theta(s_t),a_{t+1}=\pi_\theta(s_{t+1})$，属于同策略（on-policy）DPG 算法。相对的还有异策略（off-policy）DPG 算法，使用另一个策略 $\beta(s)$ 进行采样，$\beta(s)\neq\pi_\theta(s)$，一般的做法是在被改进策略 $\pi_\theta(s)$ 上增加一定的噪声，其梯度为：

$$\begin{aligned}\nabla_\theta J(\pi_\theta) &= \int_S \rho^\beta(s)\ \nabla_\theta\ \pi_\theta(s)\ \nabla_a\ Q^\pi(s,a)|_{a=\pi_\theta(s)}\,ds \\ &= E_{s\sim\rho^\beta}[\nabla_\theta\pi_\theta(s)\ \nabla_a\ Q^\pi(s,a)|_{a=\pi_\theta(s)}]\end{aligned} \tag{3-4}$$

异策略的 DPG 算法中，a_t 由行为策略产生，a_{t+1} 由目标策略产生，$a_t=$

$\beta(s_t), a_{t+1} = \pi_\theta(s_{t+1})$，参数更新过程如下：

$$\begin{aligned} \theta_{t+1} &= \theta_t + \alpha \bigtriangledown_\theta \pi_\theta(s_t) \bigtriangledown_a Q_\omega(s_t, a_t)|_{a=\pi_\theta(s)} \\ \omega_{t+1} &= \omega_t + \alpha\delta_t \bigtriangledown_\omega Q_\omega(s_t, a_t) \end{aligned} \tag{3-5}$$

3.1.4.2 DDPG 算法

深度确定性策略梯度（Deep Deterministic Policy Gradient，简称 DDPG）算法是 DPG 算法与深度学习的结合，DPG 算法具有 Actor-Critic 结构，DDPG 算法的主要改进就是将其中的 Actor 和 Critic 部分用深度神经网络表示。Actor 输出策略，Critic 估算状态动作值，利用 TD 误差进行优化，同时计算梯度用于 Actor 的更新，这些操作都能够在神经网络上实现。利用深度神经网络，可以处理更加高维的状态空间以及进行更加强大的非线性映射，从而提升强化学习算法的性能。

类似深度强化学习算法 DQN 对传统的强化学习算法 Q-Learning 的改进，除了采用神经网络，DDPG 算法在 DPG 算法的基础上还做了如下改进：

1）采用目标网络进行更新

DDPG 算法具有四个网络，Actor-eval 网络、Critic-eval 网络、Actor-target 网络、Critic-target 网络。Actor-eval 网络和 Actor-target 网络的结构、输入输出的含义相同，它们都对策略进行近似，但是参数不同，Critic-eval 网络和 Critic-target 网络也是如此，它们都对值函数进行近似。Actor-eval 定义为 $\pi_\theta(s)$，Actor-target 网络定义为 $\pi_{\theta'}(s)$，Critic-eval 网络定义为 $Q_\omega(s,a)$，Critic-target 网络定义为 $Q_{\omega'}(s,a)$，其中 θ，θ'，ω，ω' 分别为四个网络的参数。

Actor-eval 网络进行策略网络参数的更新，并根据环境当前时刻状态 s_t 输出用于交互的动作 $a_t = \pi_\theta(s_t)$，Actor-target 网络根据交互返回的下一时刻状态 s_{t+1} 估计下一时刻选择的动作 $a_{t+1} = \pi_{\theta'}(s_{t+1})$，该动作用于计算目标 Q 值。根据确定性策略梯度的思想，Actor 的目标是最大化每个状态下采取动作能够获得的 Q 值，其损失函数为

$$J(\theta) = -\frac{1}{m}\sum_{t=1}^{m} Q_\omega(s_t, a_t) \tag{3-6}$$

式3-6中的 Q_ω 即为 Critic-eval 网络的输出，Critic-eval 网络用于计算当前

状态动作的 Q 值 $Q_\omega(s_t, a_t)$，而 Critic-target 网络用于计算目标 Q 值，即当前时刻的实际奖赏与下一时刻状态动作 Q 值的折扣之和 $y_t = r_t + \gamma Q_{\omega'}(s_{t+1}, a_{t+1})$。Critic-eval 网络进行值网络参数的更新，为了近似状态动作值函数，采用当前 Q 值与目标 Q 值的均方误差作为损失函数：

$$J(\omega) = \frac{1}{m}\sum_{t=1}^{m}(y_t - Q_\omega(s_t, a_t))^2 \tag{3-7}$$

Actor-target 网络和 Critic-target 网络的参数从 Actor-eval 网络和 Critic-eval 网络中获取，可以采用 DQN 中的参数调整方法，每隔一定训练步数从两个 target 网络中完全复制参数，也可以每一步都执行软更新：

$$\begin{aligned} \theta' &= \tau\theta' + (1-\tau)\theta \\ \omega' &= \tau\omega' + (1-\tau)\omega \end{aligned} \tag{3-8}$$

式3-8中的 τ 为取值比较小的更新系数，取值 0.01 或 0.001 等。采用 eval 和 target 两套网络结构的原因是，深度强化学习在训练过程中，由于一边使用 Critic 网络计算策略梯度更新 Actor 网络的参数，一边又需要对 Critic 网络进行参数的更新，这会导致 Actor 的学习过程不稳定，因此需要对两个网络构建一个稳定的拷贝版本，使梯度不会在频繁的参数更新中发生剧烈的改变。

2）采用经验回放

相比于 PG 算法，Actor-Critic 算法的效率有一定的提升，通过 Critic 估计下一步的优劣，可以做到每一步都进行一次更新，而不用收集一整轮的数据后再进行更新，然而这样仍然没有改变每一步的交互结果仅被学习一次的情况。经验回放方法指将每一步交互得到的四元组 $e =< s_t, a_t, r_t, s_{t+1} >$ 存入大小固定的经验缓冲池 $E = \{e_1, e_2, ..., e_m\}$ 中，学习过程中每一步都从经验缓冲池随机取出数量为批次大小的数据，用于神经网络训练。

经验回放操作简单，其目的是记录历史交互数据，使它们被利用更多次，提高训练效率，同时经验回放打消了相邻训练样本之间的相关性，使网络不会过度拟合到当前观测到的样本上。策略梯度算法中，为了合理估计 Q 值，需要用到当前时刻 Q 值 $Q_\omega(s_t, a_t)$ 以及估算的下一时刻 Q 值 $Q_{\omega'}(s_{t+1}, a_{t+1})$，如果使用经验缓冲池中的历史样本 $< s_t, a_t, r_t, s_{t+1} >$，该样本的当前时刻 Q 值实际应为

$Q_{\omega_{old}}(s_t, a_t)$，下一时刻动作应为 $a_{t+1} = \pi\theta_{old}(s_{t+1})$，$\pi\theta_{old}$ 和 ω_{old} 为历史时刻的网络参数。由于历史参数无法记录，且当前策略与历史策略不同，无法准确知道历史策略下的动作，因此使用经验回放会导致行为策略和被优化策略不同，这本质上是异策略的强化学习过程。节3.1.4.1中推导了异策略确定性策略梯度算法的梯度公式，这是 DDPG 算法可以引入经验回放的重要前提。

3）引入随机噪声

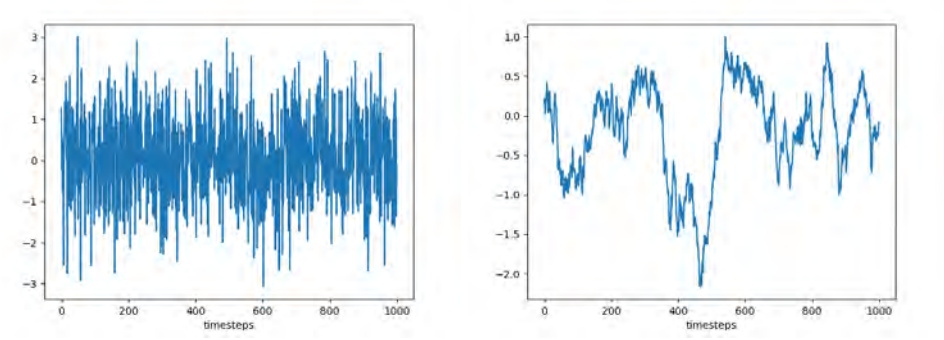
DDPG 算法在采样过程中使用 Ornstein–Uhlenbeck（简称 OU）随机过程引入随机噪声，该随机过程的公式为

$$dx_t = \theta(\mu - x_t)dt + \sigma dW \tag{3-9}$$

，在离散空间中的形式为

$$x_{t+\Delta t} - x_t = \theta(\mu - x_{t-1})\Delta t + \sigma\Delta W \tag{3-10}$$

式3-9和式3-10中 x_t 为生成的噪声值，μ 表示它的均值，θ 和 σ 为权重参数，W 为维纳过程，ΔW 是一段时间间隔内的增量，$\Delta W \sim N(0, \Delta t)$。相较于高斯噪声，OU 噪声更加适用于惯性系统的控制任务，因为 OU 噪声是自相关的，后一步的噪声受前一步的影响，如果 x_t 大于均值 μ，会呈现减小的趋势，如果 x_t 小于均值 μ，会呈现增大的趋势。这与机器人运动过程中的控制类似，机器人一般很难严格地按照指令中的数值进行运动，为了获得某一方向上稳定的速度，会在该方向或者速度大小的附近进行一定的调整，OU 噪声与这个具有惯性的调整过程类似，应用于训练过程有利于在该方向上的探索。



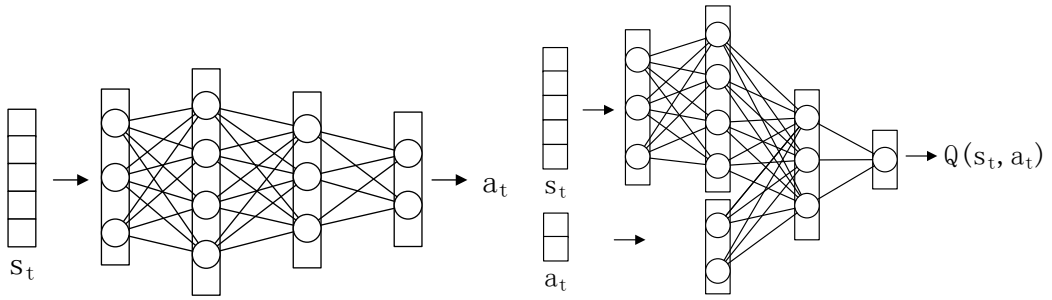
(a) 高斯噪声　　(b) OU 噪声

图 3-3: OU 噪声与高斯噪声

3.1.5 网络结构

神经网络虽然拟合能力强大，但随着网络层数的加深及每层神经元数量的增加，其计算量迅速增大。对于现阶段广为使用的各类家用机器人而言，其硬件的计算能力往往远小于个人电脑，机器人算法不仅要考虑算法性能，还要考虑计算成本，因此需要设计较轻量的网络结构。

采用深度确定性策略梯度算法，机器人导航的 Actor 网络输入为状态 s，输出为 2 维的动作 a，Critic 网络的输入为状态 s 和动作 a 的拼接，输出为对 $Q(s,a)$ 的拟合结果，因为输入维度均较低，且为了部署在成本低廉的硬件设备上，采用层数较少的全连接网络结构，本文在上述强化学习算法上设计了如表3-1所示 4 种网络结构，表中参数表示该层神经元输出向量的维度，-表示没有该层。Actor 网络采用全连接结构，如图3-4a所示。



(a) Actor 网络结构示意　　(b) Critic 网络结构示意

图 3-4: 网络结构示意图

Critic 的网络结构与 Actor 的网络结构类似，隐含层数和每个隐含层的神经元数量相同，第一个隐含层的输入为状态 s，在最后一个隐含层前加入归一化后的动作 a，具体操作为将前一层的输出向量与 a 拼接，将拼接后的值输入到下一层，如图3-4b所示。

表 3-1: 不同网络结构每层的输出维度

	网络结构 1	网络结构 2	网络结构 3	网络结构 4
隐含层 1	128	128	256	256
隐含层 2	128	128	256	256
隐含层 3	-	128	-	256
输出层	2	2	2	2

3.2 全局强化学习-PRM 规划

3.2.1 PRM

机器人传感器 FOV 是受限的，因此对于视野外的目标，机器人除了尝试靠近，没有直接的导航方法，需要借助更加高层级的环境模型。环境模型能够统一地描述导航任务的起点、终点、当前机器人位置等实际信息，从而将物理环境抽象成计算机能够处理的模型。占用栅格地图（Occupancy Grid Map）是一种常用的环境建模方法，它能够很好地处理激光雷达的输入信息。占用栅格地图以一定的尺度将现实世界划分为栅格，对于栅格 s，记录其占用率 $Odd(s) = \frac{p(s=1)}{p(s=0)}$ 表示该位置有障碍物与无障碍物概率的比值，该值越大说明 s 位置越可能有障碍物，反之越可能没有障碍物。占用栅格地图的构建是 SLAM 中的一项重要研究，在此不做赘述。

对环境建立占用栅格地图，本质上是将连续空间转化为离散空间，这样就可以使用基于图的搜索算法，在地图中搜索起点到终点之间的路径，例如使用 dijkstra、A* 等算法。这种搜索算法与其他传统的路径规划算法（例如人工势场法、单元分解法等）一样，在环境较大、障碍物较复杂的情况下，规划的计算量较大。概率路网图算法 PRM（Probabilistic Roadmaps）通过随机采样，同样进行离散空间的转化，可以有效解决大型空间和复杂条件下的路径规划问题。PRM 算法分为学习和查询两部分，学习阶段进行路网图建模，即给定地图，在空间中进行随机采样并进行 k 近邻联结，路网图构建算法如算法3.1所示；查询则是根据给定的起点和终点，给出路径，具体的做法是将起点和终点放入路网图中，使用基于图的算法搜索最短路径，让机器人沿着最短路径运动，路径上的相邻两点间的运动视为一次局部规划。

在上述算法3.1中，最为关键的一步是判断 q 到 q' 是否可达，一种简单的判断方法是，连接 q 和 q'，如果线段 qq' 上没有障碍物，则可达，反之不可达。我们称该方法为直连联结，这种判断方法存在两个问题，一是没有考虑机器人到达 q 位置的位姿和惯性，机器人不可能严格按照线段的指示直线行走，机器人在 q 位置的初始朝向和速度可能会导致运动过程中产生危险；二是按照这种构建方法，路网图会比实际情况稀疏，因为 V 中许多点对通过曲线路径可达，却

Algorithm 3.1 路网图构建

Input: n：路网图中的点数，k：每个点的最近邻连接数，Q: 包含所有点的点集
Output: 路网图 $G = (V, E)$

```
V ← ∅
E ← ∅
while |V| < n do
  repeat
    q ← Q 中的随机点
  until q 位置不会碰撞障碍物
  V ← V ∪ q
end while
for all q ∈ V do
  N_q ← V 中与 q 最近的 k 个点构成的点集
  for all q' ∈ N_q do
    if (q, q') ∉ E 且 q 到 q' 可达 then
      E ← E ∪ (q, q')
    end if
  end for
end for
```

没有加入 E 中，一定情况下会导致路网图不连通。另一种能保证路网稠密的方法是令机器人以不同位姿和初始速度在 q 出发进行局部规划到达 q'，将成功率高于阈值的点对视为两者可达，我们称该方法为访问联结，这种方法虽然保证了路网可靠，但需要花费大量时间在路网建立上。

3.2.2 基于值函数的路网联结

DDPG 算法中，Critic 网络对状态动作值函数 $Q(s,a)$ 进行拟合，Q 值的含义为当前状态 s 下采取动作 a 获得的累计折扣奖赏：

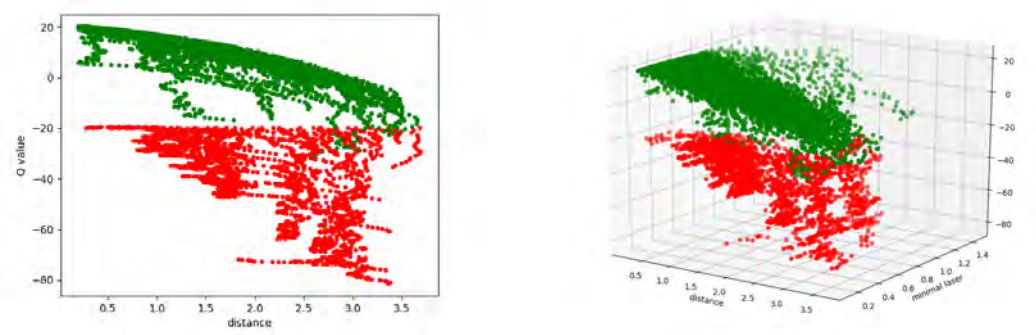
$$Q_\pi(s,a) = E_\pi[\sum_{t=0}^{+\infty} \gamma^t r_{t+1} | s_0 = s, a_0 = a] \tag{3-11}$$

无论成功还是失败，机器人导航任务都会在确定时间内结束，因此实际的累计折扣奖赏不是无限累加，Q 值为

$$Q_\pi(s,a) = E_\pi[\sum_{t=0}^{n-1} \gamma^t r_{t+1} | s_0 = s, a_0 = a] \tag{3-12}$$

式3-13中的 n 表示运行的步数，局部导航针对较近的目标，因此其取值不

会很大。由于导航成功时 agent 会接收到一个正值奖赏 R_s，而导航失败时会接收到负值奖赏 R_f，导航成功轨迹最后一项奖赏 $r_n = R_s$，导航失败轨迹上 $r_n = R_f$，因此导航成功轨迹上的状态相较失败轨迹在训练过程中会获得更高的累计折扣奖赏，也就是说，以 q' 为终点，q 位置的状态 s_q 的动作状态值越大，则 q 到 q' 可达的概率越高。



(a) Q 值与距离的关系　　(b) Q 值与距离-激光最小值的关系

图 3-5: 导航成功和失败轨迹中状态动作值的分布

上述推论在实际数据上有很强的体现，对实际Q值进行采样，得到如图3-5所示的散点图，由于导航剩余的步数难以估算，因此图3-5a采用了当前位置到终点位置的欧氏距离作为横坐标，Q 值作为纵坐标，图中上方绿色点集为成功轨迹上的 Q 值，下方红色点集为失败轨迹上的 Q 值，可以看到奖赏函数的设计导致导航成功和失败轨迹上的Q值存在明显的分界。加入激光输入的最小值后，图3-5b中成功和失败轨迹的 Q 值分界更加明显。根据该推论设计 PRM 构建算法中判断可达的值函数联结方法：对于点对 (q, q')，随机采样 m 个在 q 位置的位姿并构建状态 s_{qi}，使用策略计算动作 $a_{qi} = \pi_\theta(s_{qi})$，按式3-13计算平均联结权值 $P(q, q')$，设置联结阈值 P_{th}，若 $P(q, q') \leq P_{th}$，则认为 q 到 q' 可达，否则不可达。

$$P(q, q') = \frac{1}{m} \sum_{i=1}^{m} Q(s_{qi}, a_{qi}) \tag{3-13}$$

图3-5中长距离下的小部分导航成功状态的 Q 值混杂在导航失败状态的 Q 值点集中，理论上会导致上述联结算法将可达的两点判定为不可达，这会使PRM中相距较远的两点无法联结，而不会将本不能联结的点相连，因此影响可以忽略。受模型的拟合能力影响，依然会有少量需要花费较长时间运行的或错误的连边，实际使用过程中，在长时间无法完成局部规划时排除当前所在边，辅以行

为恢复的方法，可以不受这些边的影响。

3.3 仿真环境与实验分析

为了验证本章提出的强化学习路径规划算法性能，我们在多种环境下进行仿真实验。具体的来说，首先，为了提升训练的收敛速度，我们设计并实现了一套专用于机器人二维路径规划的仿真系统，包括障碍物地图、激光雷达、运动模型、可视化等组件，接着展示强化学习局部规划器在不同场景、不同奖赏参数下的导航成功率、速度、安全性等指标，最后结合强化学习-PRM 全局规划器，展示了本算法在较大场景下进行长距离规划的效果。

3.3.1 定制化仿真环境

目前比较主流的机器人仿真环境有 Gazebo、V-REP、MuJoCo 等。Gazebo 是开源的机器人仿真平台，具有很强的三维环境仿真、动力学仿真、传感器仿真功能，支持多种机器人模型，同时 Gazebo 依托了时下最具影响力的机器人操作系统 (Robot Operating System, 简称 ROS)，ROS 系统通用、高效且拥有活跃的开发者社区，因此广泛应用于工业生产和科学研究中。V-REP 具有强大的物理引擎和集成开发环境，仿真程度很高，支持多机器人仿真，具有很好的稳定性和交互体验，但 V-REP 部分开源。MuJoCo 是一款 2D/3D 机器人仿真环境，该平台侧重于控制和优化，广泛应用于 OpenAI 的强化学习平台 Gym，在许多强化学习研究中都有 MuJoCo 的身影。作为路径规划任务的仿真环境，Gazebo 具有很大的优势，Gazebo 的界面如图3-6所示。

仿真环境可以对时间加速，仿真速率具体指同等时间下，仿真环境中度过的时间与真实世界时间的比值。经过实验后发现，Gazebo 在高仿真速率下机器人运动会出现不可控的偏移，因为仿真平台需要对机器人各组件进行协调，在高仿真速率下，各组件更新频率以及运动指令发送频率变高，会产生不同步的情况，能够达到的最高仿真速率受计算机硬件限制。

强化学习对训练样本数量的要求极高，低仿真速率下收集数据慢，算法运行时间长，因此本文构建了一套仅用于二维路径规划的仿真系统，该系统具有如下组件和特性：

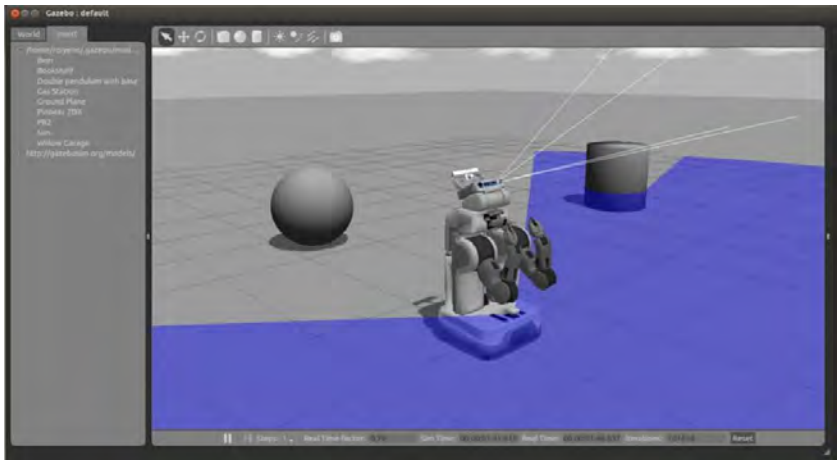


图 3-6: Gazebo 的仿真界面

1. 障碍物地图：给定一张二值图像，生成栅格地图，并进行膨胀等预处理操作，图像中黑色部分视为障碍物，白色部分视为空地。除了事先配置的静态障碍物，导航过程中还会按照随机规则动态生成一定的障碍物。机器人根据配置被视为一个规则多边形或圆形，每轮导航开始给定起点和终点坐标，机器人碰到障碍物则导航失败。
2. 激光雷达：模拟全向单线激光雷达进行 360° 扫描，以指定频率给出与最近障碍物的距离，模拟激光雷达的最小视距和最大视距等参数可自主设置。同时为了模拟真实场景中激光的精度，数据可带有噪声。
3. 运动模型：运动模型主要进行航迹推演，即根据机器人当前位姿和运动指令计算下一时刻的位姿，实际运动中，若线速度和角速度不为 0，运动轨迹可能为任意轨迹，因此将一段时间 t 采用微分思想，拆为多段 Δt 时间，使用在运行速度上优化后的切线模型进行航迹推演。切线模型假设机器人先沿着原方向行走 Δs，再转过 $\Delta\theta$ 角度，Δs 和 $\Delta\theta$ 通过牛顿力学公式分多种情况进行计算，在此不多赘述，最终航迹推演模型如式3-14所示：

$$\begin{aligned} x' &= x + \Delta s \cos\theta \\ y' &= y + \Delta s \sin\theta \\ \theta' &= \theta' + \Delta\theta \end{aligned} \tag{3-14}$$

4. 可视化界面：该平台可视化如图3-7所示，图中的黑色区域为障碍物，白色区域为空地，放射状线条为激光束，其中浅色激光束描述机器人当前朝向，地图空地上的深色线条为机器人运动路径，红色点为导航目标位

置，黄色点为随机产生的动态障碍物，随着导航的进行，机器人和动态障碍物会进行移动。

图 3-7: 仿真系统的可视化界面

5. OpenAI Gym 支持：该仿真环境与强化学习平台 OpenAI Gym 结合，为学习算法提供基础。一次导航任务视为一轮（episode），每次发送运动指令（action）为一步（step），交互后仿真环境给出奖赏、环境状态等信息供强化学习算法进行学习，给出的状态信息与地图无关，保证算法的无地图特性。

仿真实验在 CPU 型号为 Intel Core i5-7500 的台式机上运行，经过调优，不同情况下的仿真速率如表3-2所示，与此同时，Gazebo 在该硬件配置下能够稳定运行的最高仿真速率为 6x（表示模拟速度为真实世界的 6 倍），模型训练情况下为 5x，可见该定制化仿真环境具有非常大的优势。

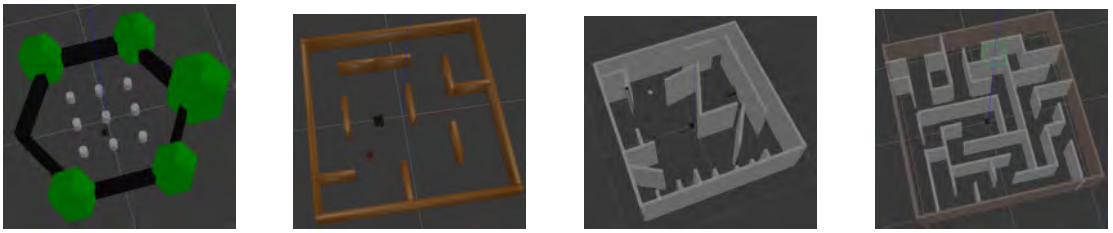
表 3-2: 仿真系统的仿真速率

运行场景	随机策略	模型推理	模型训练
无动态障碍物	625x	417x	137x
1 个动态障碍物	139x	109x	62x
3 个动态障碍物	126x	104x	57x

3.3.2 实验分析

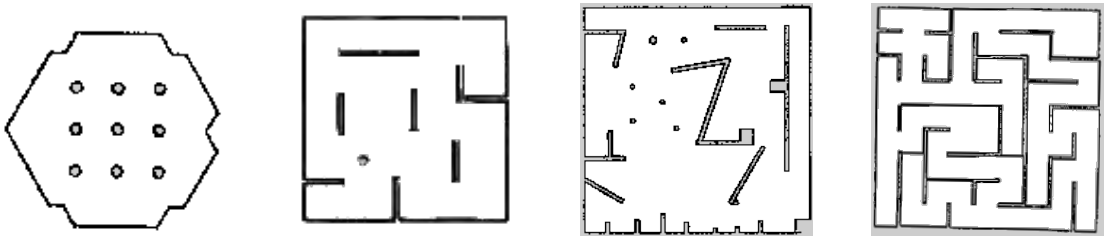
为了验证导航算法，我们搭建了 4 个仿真场景，这 4 个场景的三维立体视图如图3-8所示，导航前使用 gmapping 算法[55] 进行同步定位与建图（SLAM），并对绘制的地图进行旋转、裁剪等预处理操作，保证可视化效果，处理好的地图如图3-9所示。从场景 1 到场景 4 机器人可运动面积不断增大，障碍物复杂程度

也增大，前三个场景用于局部规划实验，场景 4 用于全局规划实验。场景 4 导航难度最高，在该迷宫环境中隔墙设置导航起点和终点或构成“U”型结构，如果其中不存在较短的可达路径，可以认为仅使用机器人传感器 FOV 内信息是无法导航的。



(a) 场景 1（6m*6m） (b) 场景 2（6m*6m） (c) 场景 3（8m*8m） (d) 场景 4（11m*11m）

图 3-8: Gazebo 中三维实验场景展示



(a) 场景 1 地图 (b) 场景 2 地图 (c) 场景 3 地图 (d) 场景 4 地图

图 3-9: 实验场景二维地图

实验对型号为 TurtleBot3-Waffle 的机器人进行仿真，TurtleBot3-Waffle 是一款广泛用于科研和教育的开源机器人，它体型小，成本低，扩展性强，配备 360° 激光雷达，主要的激光参数、运动参数及对应仿真值如表3-3所示，其硬件配置将在第5章的实际导航系统中具体介绍。为了减少运动仿真带来的误差，实验中稍微降低了最大线速度和最大角速度，为了完全同步激光雷达数据，控制指令的发送频率为 2.5Hz。

训练和测试阶段，每轮导航在地图中随机选取两个位姿作为起始状态和目标状态，并在速度范围内随机指定机器人的初始线速度和角速度，当机器人中心与目标位置距离小于 5cm 视为导航成功，机器人碰撞障碍物或在规定时间内未完成导航视为导航失败。导航实验部分的主要指标为导航成功率和导航时间，两者均取导航 500 次的平均值。实验中的 DDPG 算法均采用同一套超参数：累计奖赏参数 $\gamma = 0.99$，训练批大小 batch size 取 128，Actor 学习率为 0.0001，Critic 学习率为 0.001，经验样本池大小为 1000000，eval 网络从 target 网络更新参数时

表 3-3: 仿真参数的参考值与仿真值

参数名	参考值	仿真值
激光雷达扫描频率	5Hz	5Hz
激光雷达最小视距	0.2m	0.2m
激光雷达最大视距	3.5m	3.5m
最大线速度	0.26m/s	0.2m/s
最大角速度	1.82rad/s	1.0rad/s
线加速度	0.3m/s^2	0.3m/s^2
角加速度	2.5rad/s^2	2.5rad/s^2
控制指令发送频率	-	2.5Hz

软更新参数 $\tau = 0.001$，OU 噪声参数 $\theta = 0.15, \sigma = 0.01, \Delta t = 0.01$，同时每次训练固定随机数种子。

选用同一套奖赏函数参数和训练环境，针对表3-1中的几种不同网络结构进行实验，分析不同网络结构对训练效果的影响。在场景 1 中的导航成功率和平均耗时如表3-4所示，训练过程中采样数量和模型的表现如图3-10所示，横轴为训练的采样数，纵轴为训练阶段最优的模型表现。可见网络结构 3 下训练的模型具有较好的表现，同时训练速度和收敛速度也得到了保证，因此选用网络结构 3 进行接下来的实验。

表 3-4: 不同网络结构下的训练效果

	网络结构 1	网络结构 2	网络结构 3	网络结构 4
导航成功率	94.0%	93.0%	95.8%	95.2%
平均耗时	12.94s	13.23s	12.70s	13.23s
训练速度	60fps	50fps	55fps	45fps

奖赏函数参数影响学习算法的收敛速度、导航性能，下面对奖赏函数参数的取值进行分析。奖赏主要分为三部分：导航成功的奖励、前往目标的时间惩罚、导航失败的惩罚，式3-1中 R_s、R_m、R_f 为奖赏函数的参数。通过网格搜索进行参数选择，选取如表3-5所示的三组具有代表性的参数进行分析，每组参数在场景 2 中训练得到模型的表现如表3-6所示，每组参数在场景 2 中训练过程获得奖赏的收敛情况如图3-11所示。

表3-5中的参数组 1 是一组比较均衡的参数，在所有参数组合中具有最高的

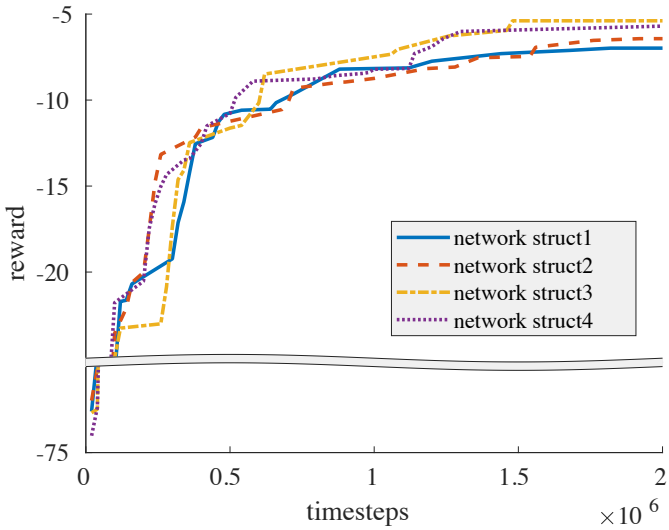


图 3-10: 不同网络结构下的训练情况

表 3-5: 各组奖赏函数参数值

参数名	参数组 1	参数组 2	参数组 3
R_s	20	20	20
R_m	-0.3	-0.8	-0.01
R_f	-20	-20	-20

导航成功率，相较于参数组 3 需要的收敛步数也较少。奖赏参数大幅改动会导致截然不同的策略，时间惩罚系数 R_m 设置的目的是对未到达终点的所有状态进行惩罚，从而引导机器人快速到达终点，但参数组 2 中过大的时间惩罚导致机器人直接撞向最近的障碍物，因为在该奖赏函数下，导航策略为快速结束任务，每在环境中停留一步都会受到巨大的惩罚，对于距离较远的目标，尽快获得失败惩罚 R_f 的总奖赏高于导航成功获得的高额时间惩罚与成功奖赏之和。

导航失败惩罚 R_f 设置的目的是对导致失败的决策进行惩罚，从而训练出更安全的策略，但是参数组 3 中失败惩罚相比时间惩罚 R_m 过大，导致机器人在训练初期的探索过程中不敢通过狭窄的路口，多次原地打转，因超时结束导航，收集到的正奖赏过少，产生图3-11中算法收敛速度慢的情况。

表 3-6: 各组参数下模型的表现

	参数组 1	参数组 2	参数组 3
导航成功率	93.6%	55.0%	91.2%
平均成功导航时间	14.28s	10.30s	15.80s
收敛步数	7.0×10^5	6.8×10^5	13.8×10^5

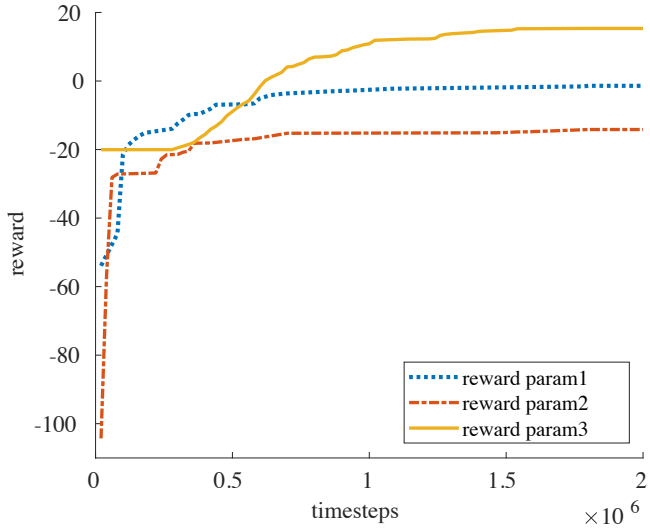


图 3-11: 不同奖赏函数下的训练情况

接下来将 RL 局部规划与 DWA 算法进行对比分析。表3-7展示了在不同场景和不同动态障碍物情况下，本章提出的局部强化学习规划算法与基于采样和估价的 DWA 算法性能对比。从表中展示的导航成功率来看，局部强化学习规划的效果比 DWA 算法更优。从实际可视化运行的效果来看，RL 局部规划已经能够处理绝大多数短距离规划，表中场景 2 和场景 3 下的成功率比场景 1 中的低，是因为这两个场景本身有许多机器人难以通行的狭窄通道或路口，动态障碍物也会阻挡机器人，使其卡在墙角或多个障碍物中，导致导航失败。

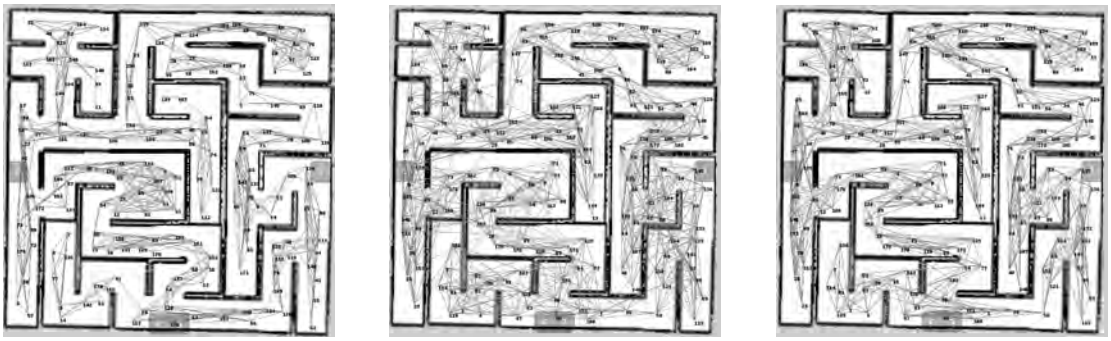
表 3-7: 动态障碍物场景下 RL 局部规划导航成功率与 DWA 算法对比

动态障碍物数量		0 个	1 个	3 个
场景 1	RL	99.5%	98.1%	94.1%
	DWA	96.7%	96.2%	93.5%
场景 2	RL	84.6%	80.8%	72.5%
	DWA	65.5%	61.0%	51.8%
场景 3	RL	61.8%	59.6%	56.6%
	DWA	41.3%	39.0%	35.4%

实验中 DWA 算法在线速度上进行 10 次采样，在角速度上进行 20 次采样，估价函数采用 obstacle cost 障碍物估价和 goal cost 目标估价，并设计了对应的估价系数，最终的估价函数形式为

$$cost = 0.2 * obstacle\_cost + 0.5 * goal\_cost \tag{3-15}$$

接下来在上述场景 4 中进行强化学习-PRM 全局规划算法验证。对场景 4 的地图建立 PRM，图3-12中从左到右分别为直连联结、访问联结和值函数联结的路网图效果，导航性能如表3-8所示。建立 PRM 时设置 k 值为 12，访问联结时进行 20 次访问，成功率超过 85% 则进行联结。PRM 建图分为采样点集和建边两步，为了公平对比，三种建图方法采用同一组点集，不同的方法进行建边，点集以每平方米 1.5 个点的密度进行采集，实验中较高的点密度会导致建图耗时更长，较少的点密度则更容易导致直接连接方法得到的路网图不连通。



(a) 直连联结 (b) 访问联结 (c) 值函数联结

图 3-12: 不同方法构建的路网图展示

从 PRM 图直观地来看，同样的点集下直接连接的路网图最稀疏，访问联结的图最稠密，值函数联结则适中。一般来说，路网图越稠密长距离规划找到的全局路径总长度就越短，因此访问联结是最优的，这与表3-8中的平均耗时效果一致。与直连联结相比，值函数联结避免了 PRM 稀疏导致的不连通问题，也提升了导航速度；与访问联结相比，值函数联结在较短的建图耗时内获得了较高的导航成功率。

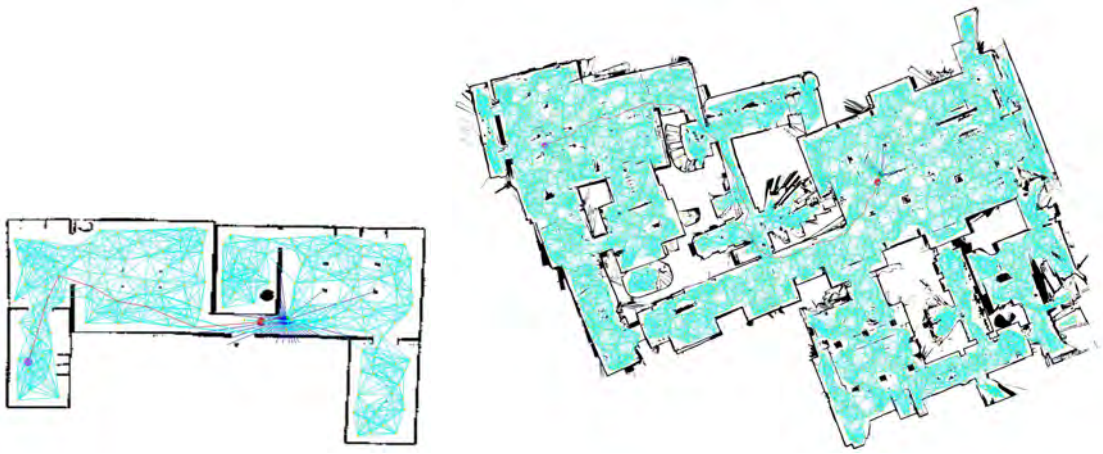
表 3-8: 不同方法构建的路网图导航性能

方法	建图耗时	导航成功率	导航平均耗时
直连联结	30 秒	联通 92.5%/不连通 53.0%	74.1s
访问联结	604 分	85.0%	68.0s
值函数联结	641 秒	91.6%	72.8s

图3-13进一步展示了值函数联结建立的路网图效果，其中图3-13a为 ROS 平台 Turtlebot 软件包中的 house 示例场景，路网图中共有 140 个点，图3-13b为由 Deutsches Museum 数据集[①]构建的场景，路网图中共有 2602 个点。图中黑色块

[①]https://google-cartographer-ros.readthedocs.io/en/latest/data.html

为障碍物，浅蓝色线条在可行区域中构成路网图，深蓝色激光束的中心为当前机器人位置，紫色点为全局导航目标，红色线条为全局规划给出的全局路径，红色点为全局路径上的局部目标。可见，在大规模场景中，本章提出的分层算法有效地给出了全局路径，并利用强化学习局部规划器沿着路径进行运动。



(a) house 场景（41m*23m） (b) Deutsches Museum 场景（70m*50m）

图 3-13: 大规模场景的导航展示

3.4 本章小结

本章提出的方法将机器人导航问题按分层结构划分为局部规划和全局规划。首先对局部规划进行马尔可夫决策过程建模，进行状态空间、动作空间和奖赏函数的设计，从而为局部规划带来端到端和无地图的特性。同时，为了应对连续动作空间和减少采样量，我们使用 DDPG 算法进行训练，进而讨论了 DDPG 算法的原理、实现和优势。其次，采用基于采样的概率路网图算法能够减少全局路径搜索在复杂环境下的计算量，为了优化概率路网建图中稀疏、耗时的问题，我们提出了基于值函数的联结方法，从而在较短时间内建立可用的路网图。最后，构建了定制化仿真环境并通过实验进行算法验证，有力地证明了我们的算法具有较优的执行效率和导航性能。

第四章　局部强化学习规划迁移研究

在第三章中，我们完整设计了一套基于强化学习的机器人导航算法。该方法中的局部规划算法具有端到端的架构，即使不依靠全局规划，作为独立的导航组件，也能应对简单场景下大部分短距离规划问题。但是由于强化学习算法存在一定的过拟合现象，算法无法应对训练过程中没有出现过的数据，因此该算法的泛化性能受到考验。为此，本章分析了增强强化学习算法泛化能力的方法，同时结合迁移学习，提出了对具有相似状态空间和动作空间的不同任务进行策略迁移的算法，并进行机器人导航策略的迁移。

4.1　深度强化学习的泛化能力

机器学习是一个庞大的研究体系，其中大多数算法的目的是根据给定的一系列训练数据，针对优化目标训练一个模型，该模型在训练数据上能取得较好的优化结果。机器学习算法的泛化性能通常是指该算法训练完成后，在训练期间从未出现过的数据上能否表现出色。一个模型的泛化性能强，说明它更好地学得了隐含在数据背后的规律，因此它对于同一规律下的其他数据，能够给出较好的输出。对于能够收敛的算法，如果其泛化性能弱，说明其存在欠拟合或过拟合的问题。

过拟合是许多机器学习算法面对的难题，如图4-1中右图所示，模型虽然在训练样本上表现良好，但考虑到数据整体的分布，表现却非常糟糕，过拟合的模型无法很好地对同分布下的其他数据进行预测。

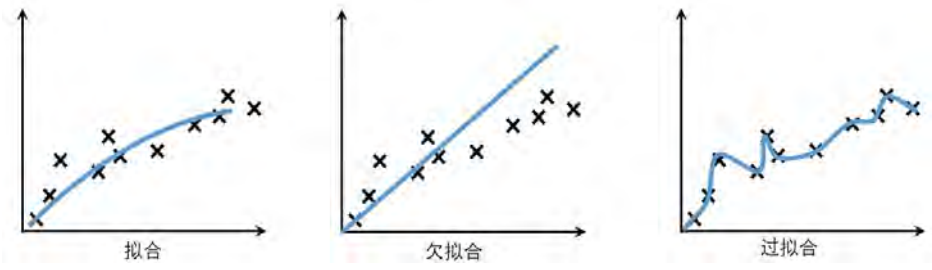


图 4-1: 模型的拟合、欠拟合和过拟合示意图

与许多机器学习方法一样，深度强化学习也面临过拟合的问题。强化学习的

任务通常被描述为马尔可夫决策过程，一个 MDP 由状态、动作、转移函数、奖赏函数构成，即 $E =< S, A, P, R >$，令 agent 在一组 MDP 上进行训练，它在其他相近 MDP 上的表现就能用来评估强化学习算法的泛化性能。训练使用的 MDP 和测试使用的 MDP 存在的差异是强化学习泛化性能面临挑战的主要因素，可能存在的差异如下：

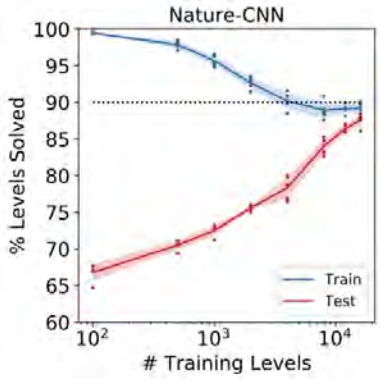
1. MDP 具有相同的转移函数，但状态有所不同。例如机器人导航问题中，将处于狭窄环境中的机器人放入宽阔环境中，传感器接收到的数据会产生变化，但策略是不变的。
2. MDP 的状态有所相似，但转移函数不同。令机器人完成复杂的运动操作时，相关参数（例如摩擦系数或传感器安装的位置等）发生变化，此时获取环境的状态不变，但采取同一个动作后，下一步状态可能不同。
3. MDP 的各项都不同，但具有相同的内在逻辑，可以将策略进行推广。例如控制机器人在平地上运行，和控制无人机进行空中运动，虽然环境从二维变为三维，但两者进行的避障这一核心操作是有内在关联的。

OpenAI 制作了一款名为 CoinRun 的游戏环境，游戏的任务是在一系列平台上跳跃，收集藏在某处的硬币，这个游戏可以无限生成结构不同的关卡，使得每一关游戏都是全新的。在该游戏上进行实验，选择固定的若干个关卡作为训练集，以深度强化学习算法在所有关卡上的平均通过率作为泛化性能指标。如图4-2中的实验结果表明，随着训练集中关卡的增加，强化学习算法在训练集上的表现越来越差，而泛化性能越来越强，即使选用 16000 个关卡进行训练，仍存在平均关卡通过率远低于训练集上通过率的情况。这说明在训练环境受限的情况下，深度强化学习算法存在很大程度的过拟合。

为了提升局部导航器的泛化性能，可以增加用于训练的 MDP 数量，虽然在无限制的环境上进行训练是不可行的，但可以将已有的环境整合在一起训练，最大程度增强泛化性能。

同时，数据增强作为传统机器学习中防止过拟合的重要手段，也可以用于强化学习。在强化学习导航上，主要通过随机化对数据进行增强。除了训练 DDPG 算法时引入的 OU 噪声，还在可以在传感器和运动模型加入随机化噪声。现实场景中，激光传感器的输入和机器人位姿信息中存在无法量化的噪声，可能是硬件设备自身带来的噪声，也可能是信息处理、定位算法中产生的噪声。因此加

(a) 游戏环境



(b) 实验结果

图 4-2: CoinRun 上的泛化性能实验

入随机化噪声进行数据增强，同时也是对真实场景的有效模拟。

4.2 深度强化学习的策略迁移

4.2.1 迁移学习和强化学习结合

尽管增强强化学习模型的泛化性能可以使模型有效适应与训练 MDP 相似的其他 MDP，但如果训练的 MDP 与测试的 MDP 差异过大，例如状态空间中各维数据的值产生较大偏移，模型将无法应对这种改变，又例如状态空间的维度发生变化，新状态空间中的状态值将无法输入维度不匹配的模型中。很多情况下，为了适应新的 MDP，需要抛弃模型中的旧参数，重新训练新参数。新模型的重新训练需要花费大量时间和算力，对于现实场景中的机器人导航这种采样代价很高的问题，重新训练模型就更加困难。因此，本节引入迁移学习，用于机器人强化学习导航在多场景切换下的重新训练，提升训练效率。

迁移学习是一项重要的机器学习研究领域，其目标是将某个任务或领域上学习到的模式和知识应用到不同但相关的问题或领域中，从而使得在新任务或领域上的学习不需要从头开始训练。当目标域的样本较少时，可以现在数据存在相关性的其他任务上学习，然后将将已学到的知识通过一定方式快速分享给新模型，从而优化和加快新模型的学习效率。根据迁移到目标域方式的不同，可以分为基于实例的迁移、基于特征的迁移、基于参数的迁移、基于关系的迁移。近年来对神经网络和深度学习的迁移学习研究众多，可以分为网络深度迁移学习、映射深度迁移学习、实例深度迁移学习和逆向的深度迁移学习[56]。

深度强化学习对训练样本量的要求较高，但在很多机器人控制任务中采集训练样本的代价较高，或受场景限制难以采集到理想的数据，此时就可以引入迁移学习，提升样本利用效率和模型的训练速度。因此将深度强化学习和迁移学习方法结合，进行强化学习策略复用和策略迁移的研究，是很有必要的。

4.2.2 基于循环一致的状态空间映射改进

将机器人导航任务看做马尔科夫决策过程 $< S, A, P, R >$，导航的目标场景进行切换后，由于机器人自身的配置和导航任务的目的不受影响，因此动作空间 A 可以认为是不变的，而状态空间 S 和转移概率 P 发生改变。奖赏函数 R 与 S 相关，同时由于转移概率隐含在转移过程中，很难估计，本文设计了基于状态空间映射的强化学习迁移算法，其基本思想如图4-3所示。

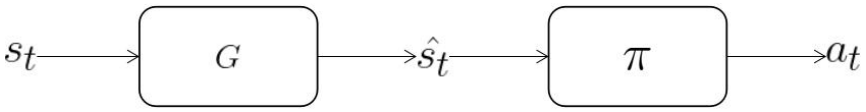


图 4-3: 状态空间映射

假设原训练场景的 MDP 为 $E' =< S', A, P', R' >$，在该 MDP 下学得策略 π，新场景的 MDP 为 $E =< S, A, P, R >$，设计一个新的映射 $G : S \rightarrow S'$，这里 G 同样采用神经网络的形式。对于 t 时刻的环境状态 s_t，通过 G 就能够实现对动作的映射：

$$a_t = \pi(G(s_t)) \tag{4-1}$$

式4-1中的 a_t 表示 t 时刻下在新场景中执行的动作，同时它也对应了原场景中 s'_t 状态下的动作。这里的 G 是一个理想映射，根据上述结构是难以训练的，最主要的原因是缺少监督数据。G 的训练需要收集数据对 (s_t, s'_t)，这类数据的收集需要人工处理，状态空间的变化无法人为模拟，那么数据对的收集将变得非常困难。

由于配对数据获取困难，本节引入基于非配对数据 (S, S') 学习映射 G 的迁移学习方法 CycleGAN[57] 及其最主要的循环一致思想。CycleGAN 是针对图像数据进行风格迁移的一种迁移学习方法，它能够实现非配对数据分布 (X, Y) 间数据样本的相互转化，CycleGAN 的大致思想如图4-4所示。称 X 为源域，Y 为目标域，假设有源域和目标域中的样本 $x \sim X, y \sim Y$，设置生成器 $G : X \rightarrow Y$ 和生成

器 $F: Y \to X$，样本 x 通过 G 映射到目标域中 $x' = G(x)$，同时 x' 又通过 F 被映射回源域中 $\hat{x} = F(x')$，理想的映射 G, F 能够实现 $x = x'$，对 y 也是一样的。根据这一条件，为了希望生成样本和原样本尽量相近，采用 L1 损失构建 CycleGAN 的循环一致损失（Cycle Consistency Loss），也称重建损失（Reconstruction Error）：

$$\begin{aligned} \mathcal{L}_{cc\_x}(G, F, X, Y) &= E_{x\sim X}[\|F(G(x)) - x\|_1] \\ \mathcal{L}_{cc\_y}(G, F, X, Y) &= E_{y\sim Y}[\|G(F(y)) - y\|_1] \end{aligned} \tag{4-2}$$

CycleGAN 是基于生成对抗网络（Generative Adversarial Networks，简称 GAN）[58] 设计的，因此对于 CycleGAN 的生成器 X 和 Y，他们都有对应的判别器 D_X 和 D_Y，判别器 D_X 的输入为源域 X 中的真实样本和通过生成器 F 产生的虚假样本，判别器的目的是将真实样本和虚假样本区分开，采用 0 和 1 的二分类损失构建其损失函数：

$$\begin{aligned} \mathcal{L}_{d\_x}(F, D_X, X, Y) &= E_{x\sim X}[\log D_X(x)] + E_{y\sim Y}[\log(1 - D_X(F(y)))] \\ \mathcal{L}_{d\_y}(G, D_Y, X, Y) &= E_{y\sim Y}[\log D_Y(y)] + E_{x\sim X}[\log(1 - D_Y(G(x)))] \end{aligned} \tag{4-3}$$

相对于判别器，生成器则要是其生成的样本尽可能真实，也就是使判别器将输出的虚假样本错误地判别为真实样本，损失函数为：

$$\begin{aligned} \mathcal{L}_{g\_x}(G, D_Y, X) &= E_{x\sim X}[\log D_Y(G(x))] \\ \mathcal{L}_{g\_y}(F, D_X, Y) &= E_{y\sim Y}[\log D_X(F(y))] \end{aligned} \tag{4-4}$$

GAN 中的生成器和判别器是相辅相成的，判别器不断地训练，要区分真假样本的能力增强的同时，生成器也根据判别其的效果，生成更难以区分的假样本。式4-3与式4-4也是相对的，最小化一者的同时就是最大化另一者。生成器与判别器一同训练，损失函数总结为：

$$\begin{aligned} \mathcal{L}_G(G, F, D_X, D_Y, X, Y) =& \mathcal{L}_{g\_x}(G, D_Y, X) + \mathcal{L}_{g\_y}(F, D_X, Y) + \\ & \mathcal{L}_{cc\_x}(G, F, X, Y) + \mathcal{L}_{cc\_y}(G, F, X, Y) \\ \mathcal{L}_D(G, F, D_X, D_Y, X, Y) =& \mathcal{L}_{d\_x}(F, D_X, X, Y) + \mathcal{L}_{d\_y}(G, D_Y, X, Y) \end{aligned} \tag{4-5}$$

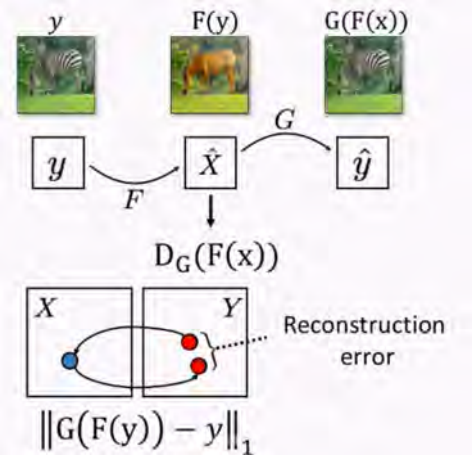


图 4-4: 循环一致损失

借助循环一致损失，可以实现非配对源域和目标域数据间的映射学习，结合循环一致损失，对强化学习的策略迁移设计如图4-5所示的基于循环一致的状态空间映射学习方法，其中 s_t, s'_t, G, F 构成了类似 CycleGAN 部分，从而有效地学习映射 G。S 和 S' 作为两个 MDP 的状态空间，状态样本可以快速高效地从环境探索的过程获取，对于 DDPG 算法，经验回放池记录了大量交互数据 $e =< s_t, a_t, r_t, s_{t+1} >$，可以取其中 s_t 构成的集合作为对 S 的近似。

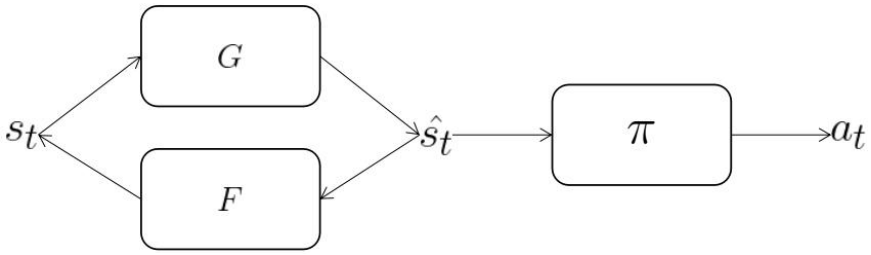


图 4-5: 基于循环一致的状态空间映射

4.3 基于单步重构的状态空间映射改进

节4.2中介绍了基于迁移学习的强化学习策略迁移方法，本节在状态空间映射思想的基础上加入时序逻辑,考虑强化学习过程中一组交互信息 $< s_t, a_t, s_{t+1} >$ 的内在关联，通过设计如图4-6所示的策略迁移框架，进一步增强策略迁移的效率。

图4-6左侧类似一个强化学习任务的基本结构，s_t 为传入策略的状态，原本由学得的策略输出动作 a_t，改由一个迁移模块（Transfer Unit）输出。迁移模块

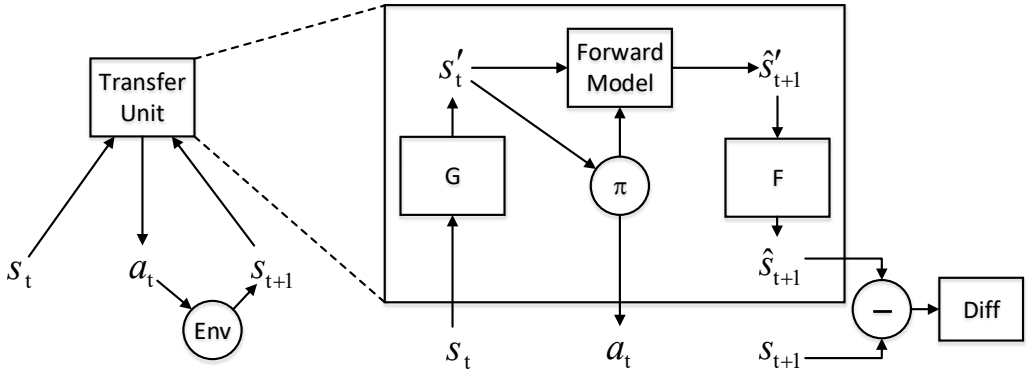


图 4-6: 基于单步重构的状态空间映射算法流程图

是策略迁移算法的核心，它有决策和学习两个阶段，决策阶段接收当前时刻状态 s_t 为输入，由状态空间映射器 G 转换为目标域下的状态 s'_t，再经已有策略 π 获得当前时刻执行的动作 a_t。学习阶段则输入当前时刻的状态 s_t 和执行动作 a_t 后的下一时刻状态 s_{t+1}，通过时序状态预测模型（Forward Model）结合循环一致损失和单步重构误差进行训练。

4.3.1 时序状态预测

为了实现时序的重构，首先引入时序状态预测模型，即图4-6中的 Forward Model。时序状态预测模型 FM 结合原 MDP 的状态 s'_t 和策略 π，对该策略下的下一时刻状态进行预测：

$$\hat{s}'_{t+1} = FM(s'_t, \pi) \tag{4-6}$$

强化学习任务依赖与环境的交互，时序状态预测的本质是学习一个基于状态空间和时间的环境表示，实现一个取代交互环境的假想模拟环境。

许多研究中都有与时序状态预测类似的思想，World Model[59] 利用变分自编码器和 MDN-RNN 实现了类似的结构，通过预测下一帧图像的分布达到了将外部模型完全放入 agent 的大脑中的目的，从而实现在抽象环境中进行完整的训练，并将这一策略转移到真实环境的效果，机器人强化学习导航的状态空间相对较小，不需要像该模型一样设计较复杂的结构。类似的思想同样也出现在基于好奇心驱动的强化学习探索[60] 中，通过计算下一时刻可能的状态衡量 agent 对整体环境的掌握程度，以掌握程度为好奇心，驱使 agent 了解所处的环境，达到探索的目的，该研究也说明了预测的有效性。该研究面向高维图像状态空间，

先对图像状态进行特征提取并在特征空间中进行预测。

在本文中，时序状态预测与 CycleGAN 结合，CycleGAN 作为风格迁移算法，一定程度上依赖原始数据的风格信息，为了不损失风格信息，在原状态空间上进行预测。同时，马尔可夫过程中蕴含了转移概率函数 P，说明一个固定状态动作对的下一个状态并不确定，在机器人导航中，由于机器人加速度和惯性没有体现在状态空间中，不确定性会更加明显，因此选用最近的 k 个历史时刻状态进行叠加，用叠加状态 $(s'_{t-k+1}, ..., s'_{t-1}, s'_t)$ 取代当前时刻状态 s'_t 作为时序状态预测模型的输入：

$$\hat{s}'_{t+1} = FM(s'_{t-k+1}, ..., s'_{t-1}, s'_t|\pi) \tag{4-7}$$

利用交互数据 $s_1, a_1, s_2, a_2, ..., s_t, a_t, s_{t+1}, a_{t+1}$ 进行训练，以 L1 范数构建损失函数：

$$\mathcal{L}_{FM}(S') = \left\|\hat{s}'_{t+1} - s'_{t+1}\right\|_1 \tag{4-8}$$

4.3.2 单步重构

时序状态预测模型给出了下一步状态可能的预测，如果让其作用于经过生成器 G 输出的状态 s'，就得到了原 MDP 中，就进一步得到了虚拟的下一时刻状态 $\hat{s}'_{t+1}$，接着再令 $\hat{s}'_{t+1}$ 经过判别器 F 回到新 MDP 的状态空间中，就得到了新任务中的下一时刻状态预测 $\hat{s}_{t+1}$，注意到该值的标签数据 s_{t+1} 是迁移模块的输入之一，因此就可以基于该预测值和标签值进行评价。定义单步重构误差（One-Step Reconstruction Error，简称 OSR）：

$$\mathcal{L}_{osr}(G, F) = \left\|F(FM(G(s)|\pi)) - s'_{t+1}\right\|_1 \tag{4-9}$$

单步重构误差是生成器 G 和 F 共同作用的结果，它考虑到强化学习作为基于时序的决策算法，在时序数据上的重构表现，一定程度上反映了 G 和 F 的映射效果。生成对抗网络中，生成器和判别器互相作用，两者的损失函数在训练过程中“此消彼长”，因此难以判断生成器的具体效果，往往需要人为查看数据进行主观评价，单步重构误差在这里也是量化生成器效果的指标。

最终，基于单步重构的状态空间映射器损失函数为

$$\begin{aligned}\mathcal{L}(G,F,D_X,D_Y,X,Y) =& \mathcal{L}_G(G,F,D_X,D_Y,X,Y)+\\ & \mathcal{L}_D(G,F,D_X,D_Y,X,Y)+\\ & \mathcal{L}_{osr}(G,F)\end{aligned} \tag{4-10}$$

迁移算法的伪代码如算法4.1所示：

Algorithm 4.1 策略迁移算法

Input: Env：新环境，Env'：旧环境，π：Env' 中训练得到的策略，n：状态样本容量，$step$：训练步数，k：用于时序预测的状态叠加数
Output: 状态空间映射 $G : S \rightarrow S'$

```
D ← ∅
随机初始化时序预测模型 FM 的网络参数
while |S'| < n do
    初始化 Env'，获得状态 s_t ← s_0
    repeat
        在 Env' 中执行动作 a ← π(s_t)
        s_{t+1} ← Env' 的当前状态
        将最近的 k 个状态叠加得到 s'
        存储状态 D ← S' ∪ s'
        采用 L_FM 和 (s_t, s_{t+1}) 训练时序预测模型 FM
        s_t ← s_{t+1}
    until s_t 为终止状态
end while
随机初始化迁移模块中 G, F, D_X, D_Y 的网络参数
i ← 0
while i < steps do
    初始化 Env，获得状态 s_0
    repeat
        在 Env 中执行动作 a ← transfer_unit(s)
        s_{t+1} ← Env 的当前状态
        随机采样 s'_t ∈ D
        采用 L_G, L_D 计算 G, F, D_X, D_Y 在 (s_t, s'_t) 上的梯度 ∇J_gen
        采用 L_osr 计算 G, F, D_X, D_Y 在 (s_t, s_{t+1}) 上的梯度 ∇J_osr
        采用 ∇J_gen + ∇J_gen 更新 G, F, D_X, D_Y 的网络权重
        s_t ← s_{t+1}
        i ← i + 1
    until s_t 为终止状态
end while
```

4.4 实验与分析

4.4.1 导航策略的泛化性能

为了评估扩充训练 MDP 数量能够带来的泛化能力，将训练场景和测试场景区别开，导航成功率如表4-1所示，表中场景 1、场景 2、场景 3 表示仅选用单一场景进行训练的训练方法，1、2 混合表示训练过程中每轮导航开始前先随机在场景 1 和场景 2 中选择，然后在选用的场景中进行导航，1、2、3 混合表示三个场景都选用。

表 4-1: 扩充训练 MDP 的泛化表现

测试 / 训练	场景 1	场景 2	场景 3
场景 1	96.2%	76.8%	39.2%
场景 2	94.8%	88.8%	44.0%
场景 3	93.4%	85.4%	58.8%
1、2 混合	96.0%	88.6%	55.0%
1、2、3 混合	95.8%	87.8%	57.6%

表4-1中，在 1、2 混合场景中训练的模型相较仅在单一场景 1 或场景 2 中训练的模型，在更复杂的场景 3 中具有最好的泛化表现，说明扩充训练 MDP 数能够增强模型的泛化能力。同时，以 1、2、3 混合训练的策略导航成功率为基准进行比较，仅在场景 1 中训练的策略在场景 1 的测试中具有最好的避障性能，高于基准线 0.4%，但在其他场景的测试中避障性能很差，在场景 2 和场景 3 中分别低 11.0% 和 18.4%，这是由模型过拟合导致的，强化学习策略在训练场景中充分且过度训练，以致于只记住了场景 1 中某些固定情况的应对策略，而忽视了应对未出现情况的整体避障能力，因此仅在训练环境中取得最好性能，这种策略在一定程度上是不合理的。类似的过拟合情况也出现在了仅在场景 2 和仅在场景 3 中训练的策略中，仅在场景 2 中训练的策略在另两个场景中较基准线低 1.0% 和 13.6%，仅在场景 3 中训练的策略在另两个场景中较基准线低 10.4% 和 2.4%，由于场景 2、场景 3 的复杂程度较场景 1 更大，因此这两个策略性能比基准低的幅度更小。

强化学习过程中，在状态和动作上增加一定的随机噪声可以增强策略的泛化性能，表4-2通过实验对比展示随机化带来的泛化性能。实验中，首先在完全准确的激光雷达数据和理想的运动模型上进行训练，接着在雷达数据的每一维叠加均值为 0、方差为 0.05 的高斯噪声，称为传感器噪声，在每一时刻的航迹推演中，机器人位姿的每一维都叠加均值为 0、方差为 0.01 的高斯噪声，称为运动噪声。训练场景为 1、2 混合，测试场景为场景 3，同时保证测试场景为无任何噪声的理想环境。导航效果对比如表4-2所示，表中增加了两种噪声的训练过程带来了最高的成功率和最低的导航平均耗时，可见随机扰动给策略带来了一定的泛化性能。

表 4-2: 随机扰动的泛化表现

	传感器噪声	运动噪声	成功率	平均耗时
1	无	无	57.6%	20.15s
2	有	无	58.6%	20.04s
3	无	有	62.8%	21.79s
4	有	有	64.0%	19.94s

4.4.2 迁移策略对经验缓冲池的影响

以 DDPG、DQN 为代表的深度强化学习算法依赖经验样本池，在开始训练前往往需要收集足够多的样本以填充经验样本池，经验样本池对算法提出了较大的采样需求，选用 Pendulum 任务和 DDPG 算法实现策略迁移，验证上述迁移算法对训练样本量的影响。Pendulum 是控制论中经典的钟摆问题，如图4-7所示为一个固定的钟摆，钟摆受重力影响会向下方旋转，通过人为施加顺时针或逆时针的力，可以控制钟摆的位置，该问题要求长时间保持钟摆尽可能朝上。

图 4-7: Pendulum 任务

在钟摆问题中使用 DDPG 算法进行强化学习训练，当 100 轮实验中平均每 200 个时间单位下取得大于-120.0 的总奖赏时，可以认为问题得到了解决。强化学习策略训练完成后，将 Pendulum 问题环境参数中的重力加速度 g 由默认的 10.0 改成 1.63，将 $g = 10.0$ 的环境视为旧环境，$g = 1.63$ 的环境视为新环境，旧环境中的策略在新环境中性能大大下降，仅能取得-655.7 的平均奖赏。接着对模型进行重新训练和迁移，实验采用的状态叠加数 $k = 5$，旧环境的样本集 S' 采用 DDPG 算法训练过程中的经验样本池 $|S'| = 20000$，状态映射模型采用具有两个隐含层的 MLP 神经网络，每个隐含层具有 128 个神经元。图4-8展示了不同训练情况下的训练效果，图中曲线表示训练过程中策略取得的奖赏值与采样量的关系，其中 ddpg with memory 线条为重新训练模型的训练效果，为了便于对比，该训练曲线省略了用于构建经验样本池的 20000 次随机采样，transfer 线条为迁移的效果，可见迁移策略在 18000 次采样下解决了问题，而重新训练需要 32000 次采样量（不包括随机采样量）。如果不在重训练前进行随机采样以填充经验样本池，则训练结果如图4-8中的 ddpg without memory 所示，算法不能收敛到 Pendulum 问题解决所要求的平均奖赏。可见迁移策略对于使用经验样本池的强化学习算法有巨大作用，迁移策略大大减少训练采样量的同时可以跳过随机采样过程，与此同时取得与原策略相当的效果。

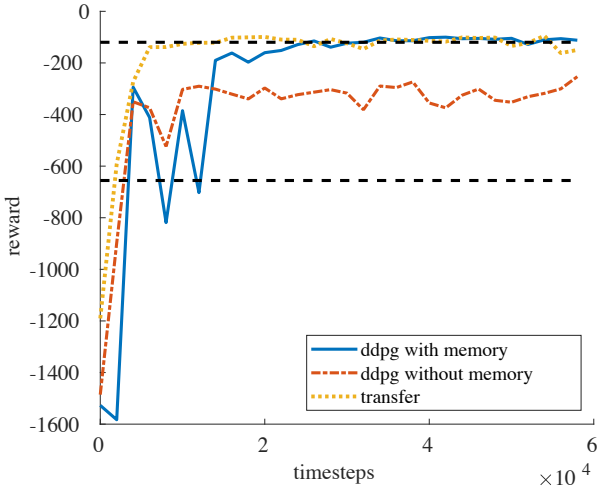


图 4-8: Pendulum 任务采样量对比

4.4.3 迁移策略在局部规划上的表现

为了验证迁移算法对局部强化学习导航规划，设置四个迁移场景进行实验，均采用在虚拟场景中训练得到的同一局部强化学习规划策略作为旧策略，实验设置如下：

1. 运动参数变更场景：机器人长时间运行后零件老化、松动可能会导致机器人的移动速率发生改变，对于可扩展性较强的机器人底盘，更换马达等也会导致运动参数发生改变。因此设置该实验场景，其中机器人的最大线速度 V_{max_l} 和最大角速度 V_{max_a} 由第3章中的 0.2m/s 和 1.0rad/s 变更为 0.15m/s 和 0.75rad/s。
2. 激光雷达变更场景：实验仿真对象 Turtlebot3-Waffle 的激光雷达模块可拆卸，因此模拟拆卸原装 HLS-LFCD2 激光雷达并安装精度更高、视距更大的 RPLIDAR-A2 激光雷达，激光雷达的最大视距由表3-3中的 3.5m 变更为 12.0m，同时将激光采样间隔由原本的 6° 变更为 5°，激光输入数据的维度 n_L 由 60 提升至 72。
3. 仿真环境变更场景：传感器在现实和各种仿真环境中具有不同的系统误差，大大制约了导航系统的性能，因此设置不同仿真环境中切换的场景，将第3章中在定制化仿真环境中的策略迁移至 Gazebo 仿真器中。
4. 虚拟到现实场景：仿真环境中训练的导航策略最终目的是应用于实际场景中，因此设置场景变换为虚拟场景到真实环境，虚拟场景使用与真实场景相同的传感器参数和运动参数，但两者的噪声分布是完成不同的。真实场景地图如图4-9所示，场景大小约为 16m*10m。

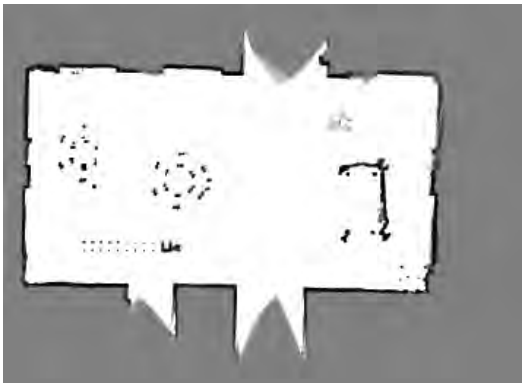


图 4-9: 测试局部规划迁移的真实场景地图

表 4-3: 局部强化学习规划的迁移表现

实验场景	策略复用	策略迁移	
	成功率	成功率	采样量
运动参数变更	70.45%	84.7%	32000(1.25%)
激光雷达变更	-	85.1%	14000(0.55%)
仿真环境变更	64.50%	82.2%	8000(0.31%)
虚拟到现实	55.0%	72.5%	6107

前三个迁移场景的实验在第3章中的场景 2 中进行，第四个迁移场景在真实环境中进行，实验采用的状态叠加数 $k = 3$，旧环境的样本大小 $|S'| = 20000$。激光雷达变更场景中，策略模型的输入维度由 64 改为 76，因此原模型无法复用，表4-3中用 “-” 标注。虚拟到现实场景中，由于真实场景收集数据慢，机器人与障碍物碰撞后需要人工介入，实验成本较高，不适合强化学习长时间的试错，因此真实场景采用离线训练，即在建图的同时收集机器人运动和传感器信息，按照迁移算法的要求将当前状态、动作、下一个状态的数据对 $d = (s_t, a_t, s_{t+1})$ 构建为离线数据集，手工控制机器人在该场景中共运行约 20 分钟，共收集 6107 条数据，Transfer Unit 使用该数据集进行训练。表4-3展示了各迁移场景下直接使用原导航策略和使用本章所述的策略迁移方法后的策略表现，原策略在 2559974 步采样下取得了 87.8% 的导航成功率，相当于在真实世界持续训练 11.85 天。在前三个迁移场景中，迁移算法分别仅在 32000 步、14000 步和 8000 步采样下就取得了 84.7%、85.1% 和 82.2% 的导航成功率，分别相当于真实世界的 3.56 小时、1.56 小时和 0.89 小时，以及重新训练采样量的 1.25%、0.55% 和 0.31%。该效果表明，使用策略迁移可以以较少的采样量在新场景中快速获得与原模型类似的导航效果。表中虚拟到现实场景也说明了本文的导航算法可以应用于实际场景，为第5章的导航系统提供了保障。

4.4.4 图像任务策略迁移

为了进一步验证本章提出的强化学习策略迁移算法及单步重构误差的作用，本节结合卷积神经网络，将机器人导航策略的迁移算法拓展至基于图像的任务中，进行进一步实验。

如图4-10，采用 Breakout-v4 任务进行图像任务策略迁移的实验，原任务是

在 atari 平台上的打砖块游戏，经过图像处理，输入为 84*84 的如图4-10a所示的二值图像输入，我们采用 PPO[19] 作为强化学习方法进行训练，训练完成的策略可以取得平均 300 以上的得分。新任务设置了如图4-10b的灰度迁移和如图4-10c的 RGB 扰动迁移，RGB 图像具有 3 通道，并在此基础上采用类似文献[61] 的方法加入若干线条进行图像扰动，旧策略在两个新任务上效果极差，平均得分均小于 2，与随机策略类似。

(a) 原任务（84*84*1） (b) 灰度迁移（84*84*1） (c) RGB 扰动迁移（84*84*3）

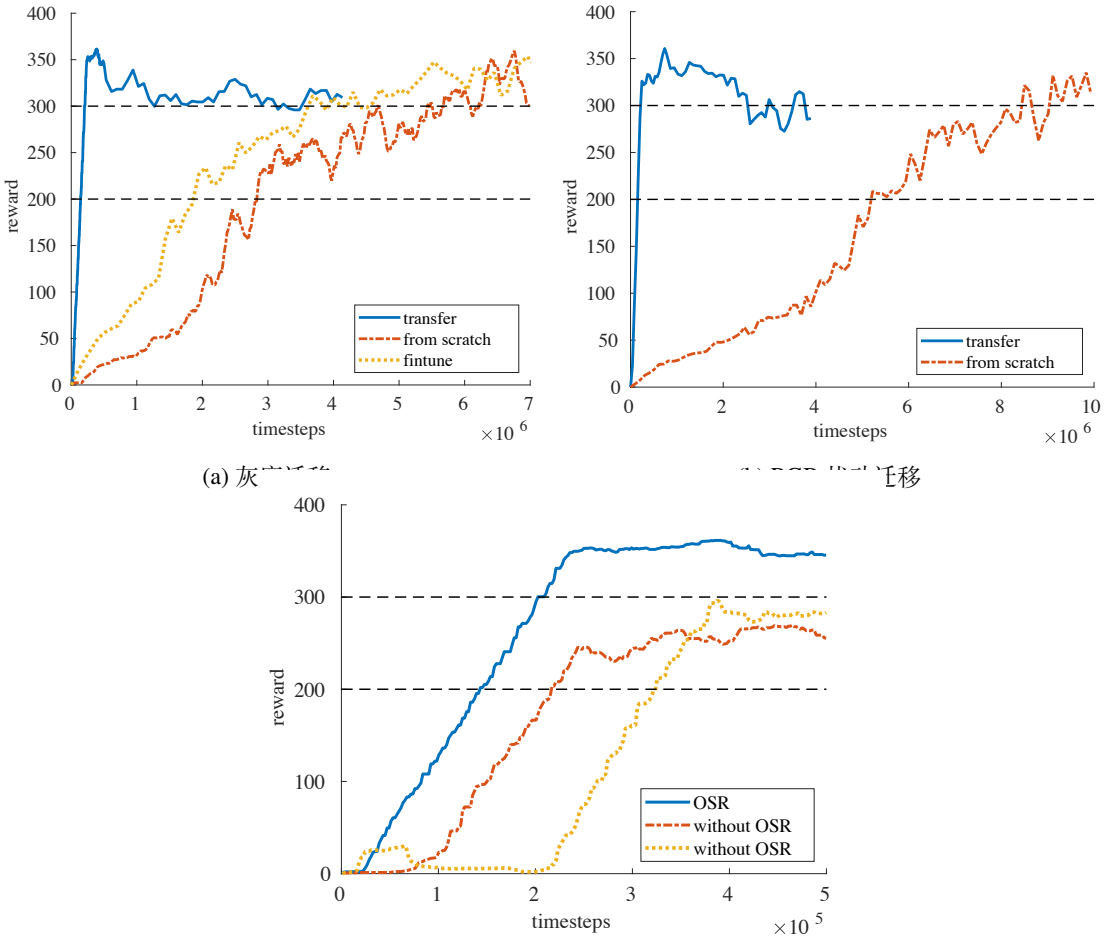
图 4-10: Breakout 迁移任务设置

表 4-4: Breakout 采样量对比

	灰度迁移		RGB 扰动迁移	
	奖赏 >200	奖赏 >300	奖赏 >200	奖赏 >300
重训练	2.847M	6.308M	5.939M	9.114M
fintune	1.946M	3.686M	-	-
OSR-transfer	0.140M	0.211M	0.164M	0.246M

图4-11a为采用重训练、fintune 和本章提出的迁移策略在灰度迁移任务上的 reward 随训练量变化情况，实验采用的状态叠加数 $k = 4$，网络采用具有 resnet 结构的卷积神经网络[57]。可见使用策略迁移算法后，强化学习算法适应新环境的速度变快，需要的采样量大大减少，表4-4中展示了奖赏达到 200 和 300 的采样量对比，采用策略迁移奖赏超过 200 仅需重训练 4.9% 和 fintune7.2% 的采样量，奖赏超过 300 仅需重训练 3.3% 和 5.7% 的采样量。图4-11b为在 RGB 扰动任务上的训练效果，该任务由于状态空间变化导致无法直接使用 fintune，但迁移策略在该任务上依然能发挥作用，奖赏达到 200 仅需重训练的 2.8% 采样量，奖赏达到 300 仅需重训练的 2.7% 采样量。

图4-11c了展示单步重构误差的作用，图中 OSR 曲线为采用单步重构误差在灰度迁移任务上的训练效果，without OSR 为两次不采用单步重构误差进行训练的效果，单步重构误差提升了最终的迁移效果，同时使训练更稳定。仅使用循环一致损失进行迁移的方法具有不稳定性，该现象在图像数据上更加明显，主要原因是生成数据的质量很难在训练过程中被量化，导致生成数据的质量非常不稳定。经过人工查看，这些训练过程中输出的图像在视觉上均没有问题，说明策略模型处理数据和人眼感知数据的方式不同。本文的迁移算法将生成的数据用于固定策略的输入，因此可以用该策略在环境中取得的总奖赏衡量具体性能，同时单步重构误差加强了对生成数据的约束，在该约束下，映射后的数据更符合策略的需求。



(a) 灰

迁移

(c) 单步重构误差效果

图 4-11: Breakout 迁移效果对比

4.5 本章小结

本章首先从强化学习的泛化性能出发，分析当前强化学习算法普遍面对的泛化性能问题，对机器人强化学习导航问题的训练过程提出了扩展训练 MDP、增加随机化等改进策略。接着针对 MDP 状态空间变化较大的新任务，引入迁移学习的方法，进而提出了基于状态空间映射的强化学习策略迁移算法。为了进一步减少迁移过程的采样量，我们提出了用于优化策略迁移的时序状态重构和单步重构误差。最后我们对本章提出的训练策略和策略迁移算法进行了验证，算法在导航任务上获得了不错的迁移效果，同时经过扩展，该方法也可用于图像状态空间的任务上。

第五章　机器人导航系统

本章主要介绍机器人导航系统，该导航系统可以作为独立的导航模块，部署于各种移动机器人上。本章的导航系统结合了本文介绍的基于强化学习的机器人导航算法及策略迁移算法，提供导航和动态避障功能，并在处理特殊情况、人机交互等方面进行了一些优化。该系统部署于 TurtleBot3-Waffle 机器人上，可以在办公场景中实地运行。

5.1　机器人导航系统背景

随着机器人技术的发展，各种各样的机器人已经出现在人们的生活、生产环境中，人们的家中有扫地机器人，商场中有导购机器人，酒店中有送餐机器人，物流中心有物流机器人等等。扫地机器人需要合理的规划从而到达家中每一个地点，同时避开障碍物，不对家具或宠物造成影响；导购机器人一般配备可视化的导购屏，当用户点击某处店铺或选定某种商品，需要快速地定位并带领用户穿过拥堵的人群前往目标；送餐机器人往往配置了若干个固定的配餐点和送餐点，不断在多点之间移动，如果位于复杂楼层中，导航过程还要考虑与电梯等设备进行协同；物流机器人同样在多地间移动，但一个物流中心通常有数十甚至数百个物流机器人，多机器人间的配合对导航模块提出挑战。无论是哪种移动机器人，如果没有外部系统辅助，机器人就需要导航模块进行自主规划移动，并且不同场景对导航有不同的要求。本章针对办公场景的需求，应用本文提出的算法，实现了一套导航系统。

5.2　机器人导航系统

5.2.1　系统需求

为了验证本文提出的基于强化学习的导航及策略迁移技术的有效性，我们在办公场景中设计了机器人导航系统。该系统的动机主要有：（1）在真实场景下

(a) 服务机器人

(b) 农业机器人

(c) 图书馆机器人

(d) 送餐机器人

图 5-1: 实际场景中的机器人

充分验证本文所提出的机器人导航技术。在现实场景中能够获得更复杂的动态障碍物以及更真实的环境噪声，根据这些真实情况能够更好地改进已有方法。(2) 该导航系统可以部署于任意对避障有较高要求的服务机器人中，提供更好的应用解决方案。

在本文实现的机器人导航系统中，主要考虑以下的需求：

1. 安全性。保证安全是导航的核心需求，主要根据避障性能判断安全性，避障分为静态避障和动态避障两方面，系统要求稳定的静态障碍物避障及有效的动态障碍物避障，由于立足于办公场景，静态障碍物主要有墙壁、挡板等，动态障碍物主要有位置不定的家具（例如椅子、垃圾桶等）以及在场景中穿梭的人。

2. 有效性。导航系统要给出合理的规划，同时高效地向目标移动，具体来说根据机器人是否已经开始移动有效性体现在两点上：机器人移动路线需在运动前的导航准备期大致规划好，此时要求系统能及时给出规划；运动开始后的导航移动期则考虑机器人导航的时间开销。

3. 可靠性。导航系统要能够自行处理部分极端情况，例如动态障碍物将机器人包围、导航失败后无法移动的情况，此时机器人需要判断自身所处的危险环境并及时上报，在可能的情况下主动走出困境。

5.2.2 系统架构

5.2.2.1 TurtleBot3-Waffle 机器人

TurtleBot 是一款移动机器人的总称，目前有三代产品，TurtleBot、TurtleBot2 及 TurtleBot3，图5-2为各代产品的实物展示图，TurtleBot3 是目前最新一代机器人。TurtleBot3 根据硬件结构分为 Turtlebot3-Burger 和 Turtlebot3-Waffle，Turtlebot3-Burger 高瘦，造型和前两代产品类似，相对而言 Turtlebot3-Waffle 则扁平，这种设计在运动时需要关注的三维障碍物信息更少，更适合部署二维导航算法。Turtlebot3-Waffle 根据硬件配置又被分为 Turtlebot3-Waffle 和 Turtlebot3-Waffle-Pi，区别在于前者搭载了 Intel joule 物联网开发板和 Intel Realsense R200 深度相机而后者搭载了树莓派 Pi 3 和树莓派相机，本系统使用经过改装的 Turtlebot3-Waffle 机器人。

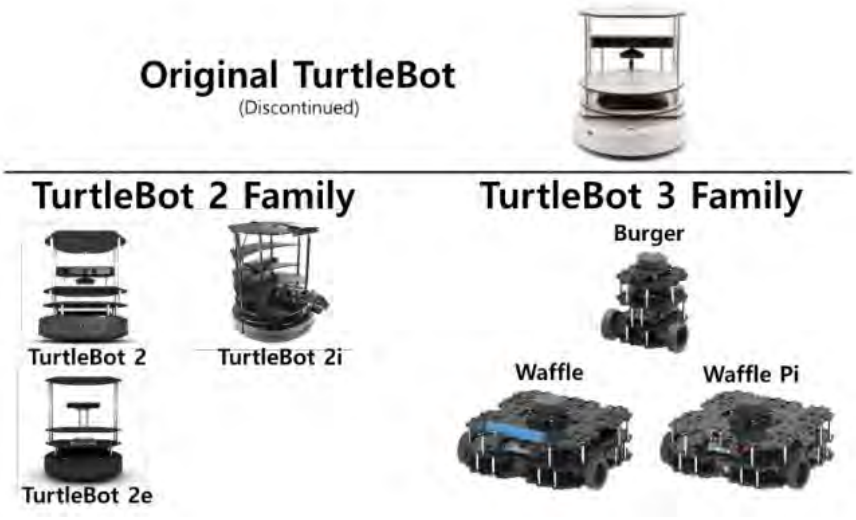


图 5-2: TurtleBot 各代产品实物图

TurtleBot 是一款可扩展性强、低成本的开源机器人，广泛用于科学研究中。TurtleBot 可以轻松地搭载机器人算法, 实现很多功能，例如机器人送餐、机器人货运、机器人教育、机器人导购等实际应用。TurtleBot 配有单线激光雷达和视觉传感器，可以轻松搭载许多基于视觉的服务，例如行人检测、人脸识别、姿态识别、物体跟随等，为机器人室内导航和其他科学研究提供了巨大的帮助。在硬件结构上，TurtleBot 具有层叠式结构，最下层为移动底盘，中间层安装控制器，上层安装传感器，其可扩展性体现在根据需求可以继续向上堆叠，放置其他设

备，例如用于交互的笔记本电脑、触摸屏等。

TurtleBot 提供了 ROS 支持，在 ROS 的仿真环境中可以导入 TurtleBot 模型，ROS 也可以部署于实体 TurtleBot 机器人上实现便捷的控制，进行硬件的实时控制。本系统的硬件清单如表5-1所示。

表 5-1: Turtlebot3-Waffle 硬件清单

智能马达	XM430
控制器	OpenCR 控制器
SBC 单板电脑	Intel Joule 物联网开发板
激光雷达	HLS-LFCD2 激光雷达
深度相机	Intel Realsense R200 深度感测摄影机
电源	2200mAh 电池

传感器方面，导航模块主要使用激光雷达作为传感器输入，深度相机用于可视化展示，激光雷达的主要参数如表5-2所示。

表 5-2: 激光雷达主要参数

尺寸	69.5(W)*95.5(D)*39.5(H)mm
检测距离	120mm~3500mm
角范围	360°
采样率	1.8kHz
扫描频率	5Hz

5.2.2.2 具体流程

导航系统的具体部署流程如图5-3所示。除了定位和导航模块，其他均为前期的预处理操作，在性能较强的个人电脑上完成。首先在实际场景中部署无线网络，便于进行机器人远程操作，连接上机器人后采用 gmapping 算法进行建图，并对建立的地图进行旋转、裁剪等处理后保存。接着按照第4章中的方法对第3章中训练完成的模型进行策略迁移，使其适应真实场景中的运动模型和传感器噪声。最后根据建图模块生成的地图按第3章中的算法建立 PRM 路网图。

真实场景的导航系统中，受动态障碍物、传感器视距、导航算法性能等因素影响，并不能保证每次导航都准确地到达终点，在机器人进入危险情况后，准

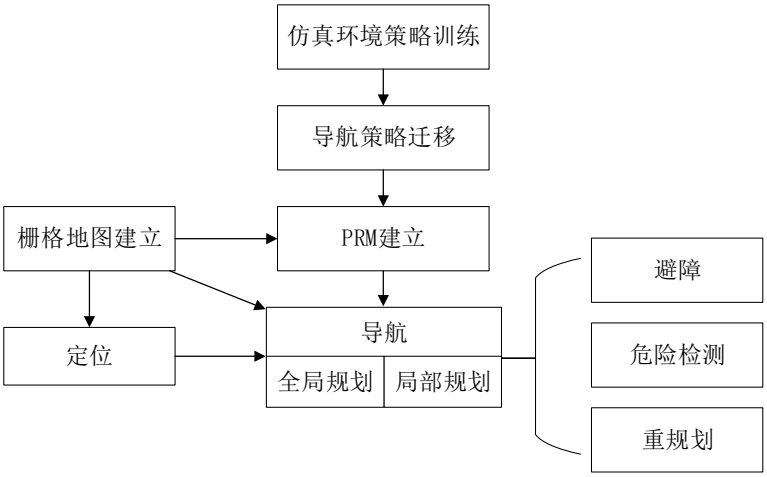


图 5-3: 导航系统部署流程

确识别并快速响应是机器人导航系统需要具备的一项功能。这项功能具有重要的意义，一方面，在导航失败的情况下及时预警，可以调整位姿重新进行规划，从而提升系统整体的导航成功率；另一方面，在受传感器视距外的障碍物阻挡导致导航失败时可以及时离开危险状态，避免人工干涉甚至需要重新部署系统的情况，从而保障导航系统的稳定运行。为此，本导航系统实现了如图5-4所示的导航任务处理流程，在导航的分层规划上增加了碰撞检测和行为恢复功能。

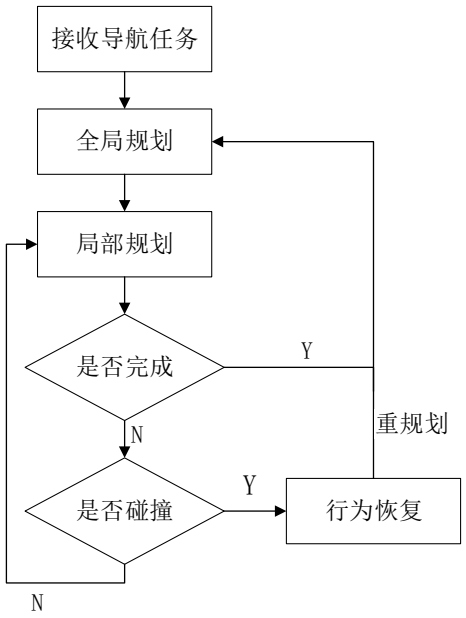


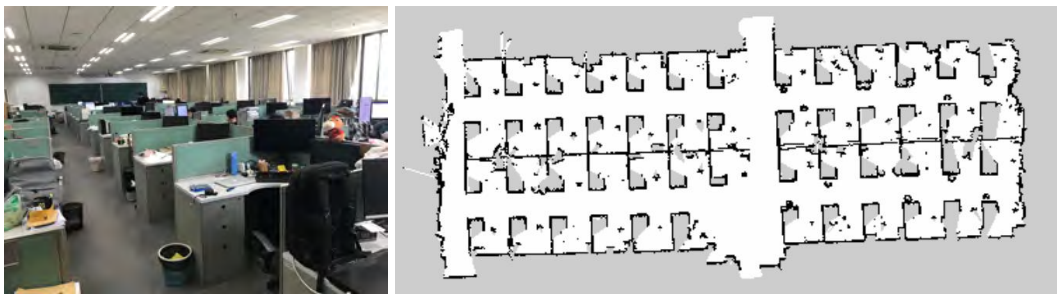
图 5-4: 导航任务的处理流程

碰撞检测利用 Turtlebot3-Waffle 机器人的深度相机输入，对机器人前进方向

上三维空间内的障碍物进行检测，由于单线激光雷达只能在其安装高度上扫描视野平面内的障碍物，无法根据其他位置的形状将在高度上不规则或高度低于激光雷达的物体识别为障碍物，容易对机器人的行动造成阻挡，因此需要其他传感器的辅助。行为恢复则在机器人长时间偏离定位系统输出的位姿时执行原地旋转、后退等的固定动作，在实际场景中定位偏离常常由人为搬运机器人或机器人导航中撞上障碍物导致，对于以前者为代表的机器人绑架问题，行为恢复能够及时初始化定位系统，同时为定位系统的重定位提供了足够的时间和数据；对于后者为代表的导航失败情况，行为恢复让机器人回到安全状态，并以该状态为初始位置重新开始导航流程。

5.2.3 系统效果

在图5-5a展示的实验办公场景中进行导航系统效果展示，该场景长 20m，宽 9m，具有 65 个工位，场景中有许多办公桌、椅子、电子设备等静态障碍物，也有行走的人、可移动的垃圾桶、形状不一的纸箱等动态障碍物。事先采用建图算法进行建图，该场景的二维地图如图5-5b所示，图中的白色区域代表可通行区，黑色区域代表静态障碍物，灰色区域为未知区域，未知区域也被认为不可通行。



(a) 现实场景 (b) 二维地图

图 5-5: 办公室场景展示

由于本文提出的导航算法尽可能地采用轻量级网络，因此对硬件的计算性能要求较低，在定位等组件开启的情况下，与采用 DWA 算法作为局部规划的方法相比，强化学习导航的运行速率约为其 8 倍，如图5-6所示。

接下来展示导航系统的实际运行效果，进行仅局部规划导航、长距离规划导航和静态障碍物变化导航三个场景的展示。图5-7为导航的起始状态，图5-7a为真实场景，图中 Turtlebot3-Waffle 机器人处于启动状态，图5-7b为 ROS 平台的

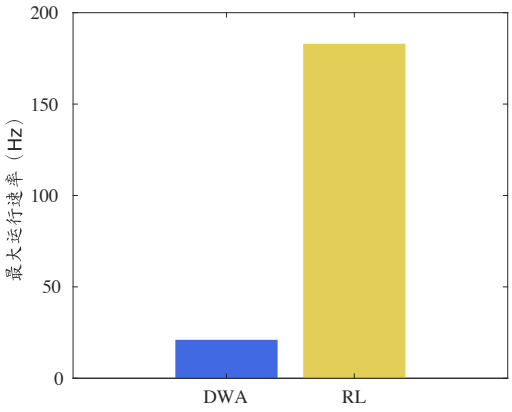
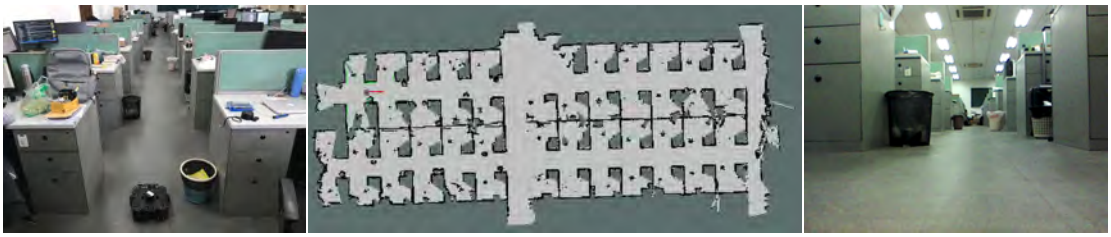


图 5-6: 导航最大运行速率

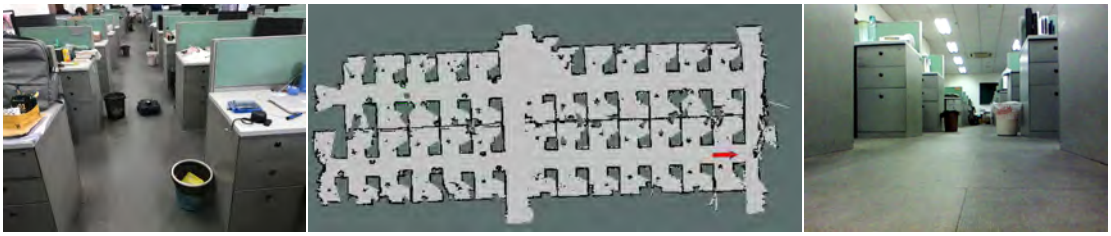
RVIZ 工具展示的二维地图可视化界面，图中机器人图标为机器人表示当前位姿，红色箭头表示导航任务的终点，绿色点集为激光传感器的输入信息，图5-7c为机器人的深度相机输入。三个场景的导航初始情况机器人都位于地图左上角，朝向右侧。



(a) 现实场景 (b) 二维地图 (c) 机器人视觉

图 5-7: 导航系统初始状态展示

图5-8展示静态避障效果。首先仅使用强化学习局部规划器进行导航，以地图右下角为导航目标，如图5-8b所示，机器人快速前进并对静态障碍物垃圾桶进行躲避，躲避过程如图5-8a所示。

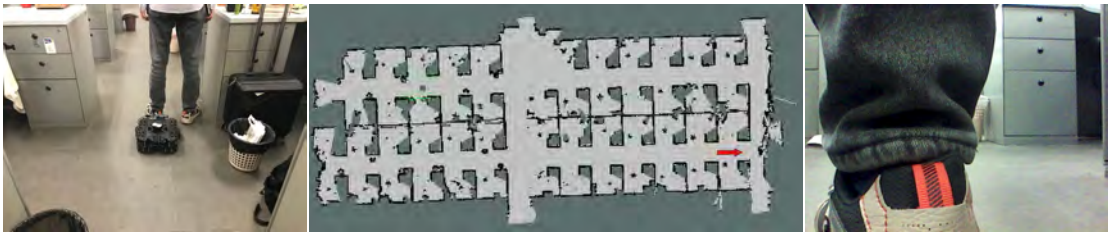


(a) 现实场景 (b) 二维地图 (c) 机器人视觉

图 5-8: 静态障碍物躲避效果

图5-9和图5-10展示动态避障效果。机器人高速运动时，在机器人行驶路线正前方突然出现一位行人，如图5-9a所示。行人出现后，机器人快速刹车减速并

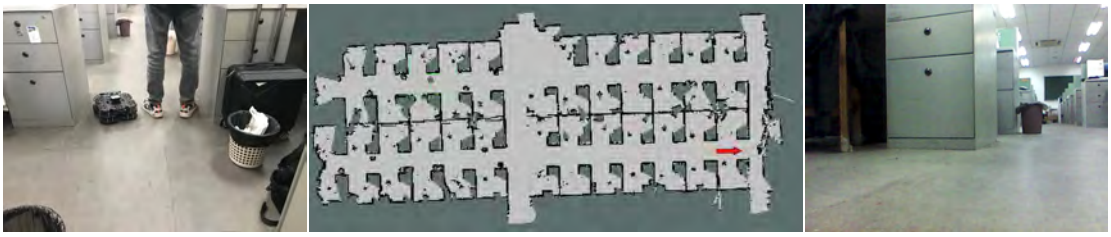
稍微后退，接着向左转向并加速前进绕开行人。



(a) 现实场景 (b) 二维地图 (c) 机器人视觉

图 5-9: 动态障碍物躲避前的情况

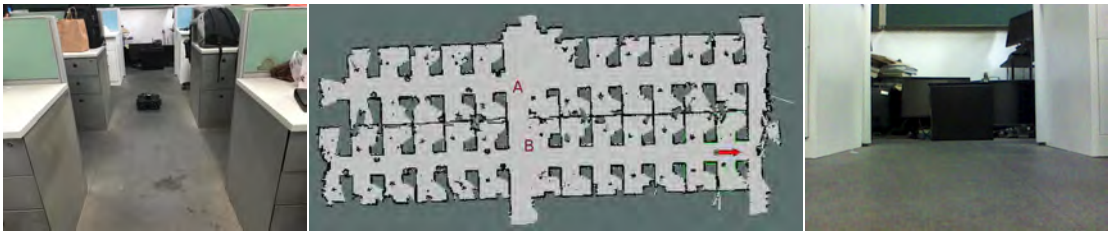
如图5-10所示为机器人左转向绕开行人的过程，此时行人左脚与左侧桌子的距离约为 50cm，绕开行人后机器人向右转向并远离左侧的桌子。



(a) 现实场景 (b) 二维地图 (c) 机器人视觉

图 5-10: 动态障碍物躲避后的情况

图5-11展示机器人完成导航的过程。机器人从左上角起点出发，直行一段后，在图5-11b地图中间的 A 点处右转，接着在 B 点处的空地逐步左转到达终点所在的过道，然后保持直行，最终机器人运动到达地图右下角的终点，完成本次导航，如图5-11a所示。



(a) 现实场景 (b) 二维地图 (c) 机器人视觉

图 5-11: 仅使用局部规划完成导航后的情况

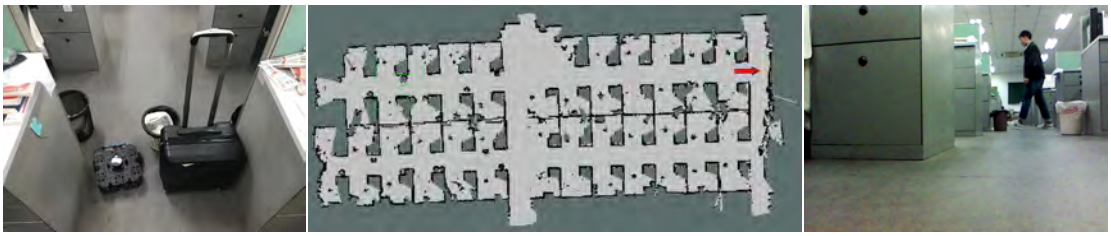
图5-12展示长距离规划效果，对办公室地图建立 RPM 路网图后，使用第3章所述的分层结构导航。与图5-7b相同，以地图左上角，朝向右侧作为起始状态，如图5-12b所示设置地图左下角某位置为终点，导航系统快速给出了全局路径并操作机器人前往终点，机器人直行一段后在地图中间的 A 点处右转，接着在 B 点

处右转，然后保持直行，最终机器人到达左下角的终点，导航完成后如图5-12a所示。

(a) 现实场景 (b) 二维地图 (c) 机器人视觉

图 5-12: 长距离规划完成导航后的情况

在实际场景中，静态障碍物也常常会发生变化，如图5-13a所示，场景中原本位于道路左侧的垃圾桶位置发生变化，同时道路右侧出现了新的障碍物，在地图中，这条通道左侧由原本的静态障碍物占据，右侧由新的动态障碍物占据，以左上角为起点，如图5-13b所示的右上角为终点，机器人在通过该通道时，从静态地图中不可通行的道路左侧无障碍物的位置通过。



(a) 现实场景 (b) 二维地图 (c) 机器人视觉

图 5-13: 静态障碍物变化后的导航效果

5.3 本章小结

本章主要介绍了基于本文提出的机器人强化学习导航算法实现的机器人导航系统。该导航系统充分验证了本文提出的导航算法以及迁移算法的实用性和有效性，结合 ROS 的可视化工具 rviz 使导航系统更易于操作。

第六章　总结与展望

6.1　总结

移动机器人正越来越多的进入各行各业，其面对的环境更加复杂、庞大，对技术的要求也越来越高，机器人自主导航技术作为当下机器人智能领域重要的研究内容，面临着新的问题和挑战。

为了优化导航的效果，本文分析了移动机器人导航的相关研究，围绕导航的安全性、可用性和可扩展性，展开了大量理论研究和应用实践，具体工作内容如下：

1. 本文对机器人导航问题的马尔科夫过程建模展开探讨，提出了基于分层结构的机器人强化学习导航算法。局部规划部分基于深度确定性策略梯度算法，其端到端、无地图的特性使得它不依赖建图算法、不依赖人工设计估价方案，连续动作空间的设计也赋予机器人更强的机动性。全局规划部分是强化学习与 PRM 的结合，为了优化路网建图中稀疏、耗时的问题，我们提出了基于值函数的联结方法，从而在较短时间内建立可靠的路网图。同时，为了高效训练和评估上述算法，我们构建了一套定制化仿真环境，在多个场景中的实验证实了算法的有效性和适用性。
2. 本文对机器人导航场景变换的问题进行研究，给出了跨场景导航的泛化性能指导，提出了基于单步重构和状态空间映射的强化学习策略迁移算法。提升模型自身的泛化性能，是在状态空间变换不大的场景中应用模型的一项重要手段，对变换较大的场景无需重新训练模型，而是基于迁移学习进行状态空间映射，同时结合单步重构误差加速迁移强化学习策略，相比于重新训练模型，迁移算法以较少的采样量得到了效果相当的模型。
3. 本文的方法成功地应用在真实场景中。结合定位、建图等算法，本文构建了一套完整的强化学习导航系统，Turtlebot3-Waffle 机器人搭载该系统，在真实环境中实现了智能规划和动态避障。

6.2 展望

为了实现更加智能的机器人控制，本文建议通过如下几个方面对深度强化学习机器人导航以及机器人智能的其他相关技术进行深入研究：

1. 考虑到机器人控制单元的硬件成本，本文没有使用深度相机的输入信息。在算力较强的机器人硬件设备上，将图像数据与激光数据结合，可以获得更加丰富的信息，从而提升机器人的环境感知能力。
2. 本文使用了深度确定性策略梯度算法进行研究，近几年深度强化学习领域不断涌现新的优秀算法，可以尝试结合其他基于策略梯度的强化学习算法或技术，使导航更加智能。
3. 相比于传统的搜索、规划、采样等手段，深度强化学习有其独特的优势，希望本文在导航问题上的深度强化学习实践可以对其他机器人技术有所启发，强化学习与机器人智能会走向更美好的明天。

参考文献

[1] ANDERSON P, CHANG A X, CHAPLOT D S, et al. On evaluation of embodied navigation agents[J/OL]. CoRR, 2018, abs/1807.06757. http://arxiv.org/abs/1807.06757.

[2] LAVALLE S M, et al. Rapidly-exploring random trees: A new tool for path planning[J]. 1998.

[3] KAVRAKI L E, LATOMBE J. Randomized preprocessing of configuration space for fast path planning[C/OL]//Proceedings of the 1994 International Conference on Robotics and Automation, San Diego, CA, USA, May 1994. IEEE Computer Society, 1994: 2138-2145. https://doi.org/10.1109/ROBOT.1994.350966.

[4] FOX D, BURGARD W, THRUN S. The dynamic window approach to collision avoidance[J/OL]. IEEE Robotics Autom. Mag., 1997, 4(1): 23-33. https://doi.org/10.1109/100.580977.

[5] 刘利强, 戴运桃, 王丽华, 等. 基于蚁群算法的水下潜器全局路径规划技术研究[J]. 系统仿真学报, 2007, 19(18): 4174-4177.

[6] RAM A, BOONE G, ARKIN R C, et al. Using genetic algorithms to learn reactive control parameters for autonomous robotic navigation[J/OL]. Adapt. Behav., 1994, 2(3): 277-305. https://doi.org/10.1177/105971239400200303.

[7] LUO C, YANG S X. A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments [J/OL]. IEEE Trans. Neural Networks, 2008, 19(7): 1279-1298. https://doi.org/10.1109/TNN.2008.2000394.

[8] TSE P, LANG S, LEUNG K, et al. Design of a navigation system for a household mobile robot using neural networks[C]//1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227): volume 3. [S.l.]: IEEE, 1998: 2151-2156.

[9] FU Y, LANG S Y T. Fuzzy logic based mobile robot area filling with vision system for indoor environments[C/OL]//Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA'99, Monterey, California, USA, November 8-9, 1999. IEEE, 1999: 326-331. https://doi.org/10.1109/CIRA.1999.810069.

[10] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[J/OL]. CoRR, 2013, abs/1312.5602. http://arxiv.org/abs/1312.5602.

[11] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J/OL]. Nat., 2015, 518(7540): 529-533. https://doi.org/10.1038/nature14236.

[12] SCHAUL T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay [C/OL]//BENGIO Y, LECUN Y. 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016. http://arxiv.org/abs/1511.05952.

[13] WANG Z, SCHAUL T, HESSEL M, et al. Dueling network architectures for deep reinforcement learning[C/OL]//BALCAN M, WEINBERGER K Q. JMLR Workshop and Conference Proceedings: volume 48 Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. JMLR.org, 2016: 1995-2003. http://proceedings.mlr.press/v48/wangf16.html.

[14] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of go with deep neural networks and tree search[J/OL]. Nat., 2016, 529(7587): 484-489. https://doi.org/10.1038/nature16961.

[15] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[J/OL]. CoRR, 2018, abs/1801.01290. http://arxiv.org/abs/1801.01290.

[16] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[C/OL]//BENGIO Y, LECUN Y. 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016. http://arxiv.org/abs/1509.02971.

[17] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning[C/OL]//BALCAN M, WEINBERGER K Q. JMLR Workshop and Conference Proceedings: volume 48 Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. JMLR.org, 2016: 1928-1937. http://proceedings.mlr.press/v48/mniha16.html.

[18] SCHULMAN J, LEVINE S, ABBEEL P, et al. Trust region policy optimization [C/OL]//BACH F R, BLEI D M. JMLR Workshop and Conference Proceedings:

volume 37 Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. JMLR.org, 2015: 1889-1897. http://proceedings.mlr.press/v37/schulman15.html.

[19] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J/OL]. CoRR, 2017, abs/1707.06347. http://arxiv.org/abs/1707.06347.

[20] STENTZ A. Optimal and efficient path planning for partially-known environments[C/OL]//Proceedings of the 1994 International Conference on Robotics and Automation, San Diego, CA, USA, May 1994. IEEE Computer Society, 1994: 3310-3317. https://doi.org/10.1109/ROBOT.1994.351061.

[21] STENTZ A. The focussed d* algorithm for real-time replanning[C/OL]// Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes. Morgan Kaufmann, 1995: 1652-1659. http://ijcai.org/Proceedings/95-2/Papers/082.pdf.

[22] DU M, CHEN J, ZHAO P, et al. An improved rrt-based motion planner for autonomous vehicle in cluttered environments[C/OL]//2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014. IEEE, 2014: 4674-4679. https://doi.org/10.1109/ICRA.2014.6907542.

[23] LAU B, SPRUNK C, BURGARD W. Kinodynamic motion planning for mobile robots using splines[C/OL]//2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, 2009, St. Louis, MO, USA. IEEE, 2009: 2427-2433. https://doi.org/10.1109/IROS.2009.5354805.

[24] BOOR V, OVERMARS M H, VAN DER STAPPEN A F. The gaussian sampling strategy for probabilistic roadmap planners[C/OL]//1999 IEEE International Conference on Robotics and Automation, Marriott Hotel, Renaissance Center, Detroit, Michigan, USA, May 10-15, 1999, Proceedings. IEEE Robotics and Automation Society, 1999: 1018-1023. https://doi.org/10.1109/ROBOT.1999.772447.

[25] HSU D, JIANG T, REIF J H, et al. The bridge test for sampling narrow passages with probabilistic roadmap planners[C/OL]//Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA 2003, September 14-19, 2003, Taipei, Taiwan. IEEE, 2003: 4420-4426. https://doi.org/10.1109/ROBOT.2003.1242285.

[26] HSU D, SÁNCHEZ-ANTE G, CHENG H, et al. Multi-level free-space dilation for sampling narrow passages in PRM planning[C/OL]//Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, May 15-19, 2006, Orlando, Florida, USA. IEEE, 2006: 1255-1260. https://doi.org/10.1109/ROBOT.2006.1641881.

[27] TAHIR Z, QURESHI A H, AYAZ Y, et al. Potentially guided bidirectionalized rrt* for fast optimal path planning in cluttered environments[J/OL]. Robotics Auton. Syst., 2018, 108: 13-27. https://doi.org/10.1016/j.robot.2018.06.013.

[28] BORENSTEIN J, KOREN Y. The vector field histogram-fast obstacle avoidance for mobile robots[J/OL]. IEEE Trans. Robotics Autom., 1991, 7(3): 278-288. https://doi.org/10.1109/70.88137.

[29] SIMMONS R G. The curvature-velocity method for local obstacle avoidance [C/OL]//Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, USA, April 22-28, 1996. IEEE, 1996: 3375-3382. https://doi.org/10.1109/ROBOT.1996.511023.

[30] ULRICH I, BORENSTEIN J. Vfh*: Local obstacle avoidance with look-ahead verification[C/OL]//Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000, April 24-28, 2000, San Francisco, CA, USA. IEEE, 2000: 2505-2511. https://doi.org/10.1109/ROBOT.2000.846405.

[31] SUTTON R S. Learning to predict by the methods of temporal differences[J/OL]. Mach. Learn., 1988, 3: 9-44. https://doi.org/10.1007/BF00115009.

[32] WATKINS C J C H, DAYAN P. Technical note q-learning[J/OL]. Mach. Learn., 1992, 8: 279-292. https://doi.org/10.1007/BF00992698.

[33] SUTTON R S, MCALLESTER D A, SINGH S P, et al. Policy gradient methods for reinforcement learning with function approximation[C/OL]//SOLLA S A, LEEN T K, MÜLLER K. Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]. The MIT Press, 1999: 1057-1063. http://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.

[34] WILLIAMS R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J/OL]. Mach. Learn., 1992, 8: 229-256. https://doi.org/10.1007/BF00992696.

[35] PETERS J, VIJAYAKUMAR S, SCHAAL S. Natural actor-critic[C/OL]// GAMA J, CAMACHO R, BRAZDIL P, et al. Lecture Notes in Computer Science: volume 3720 Machine Learning: ECML 2005, 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005, Proceedings. Springer, 2005: 280-291. https://doi.org/10.1007/11564096_29.

[36] BHATNAGAR S, SUTTON R S, GHAVAMZADEH M, et al. Incremental natural actor-critic algorithms[C/OL]//PLATT J C, KOLLER D, SINGER Y, et al. Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007. Curran Associates, Inc., 2007: 105-112. https://proceedings.neurips.cc/paper/2007/hash/6883966fd8f918a4aa29be29d2c386fb-Abstract.html.

[37] DEGRIS T, PILARSKI P M, SUTTON R S. Model-free reinforcement learning with continuous action in practice[C/OL]//American Control Conference, ACC 2012, Montreal, QC, Canada, June 27-29, 2012. IEEE, 2012: 2177-2182. http://ieeexplore.ieee.org/document/6315022/.

[38] ZHENG G, ZHANG F, ZHENG Z, et al. DRN: A deep reinforcement learning framework for news recommendation[C/OL]//CHAMPIN P, GANDON F L, LALMAS M, et al. Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018. ACM, 2018: 167-176. https://doi.org/10.1145/3178876.3185994.

[39] YU L, ZHANG W, WANG J, et al. Seqgan: Sequence generative adversarial nets with policy gradient[C/OL]//SINGH S P, MARKOVITCH S. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. AAAI Press, 2017: 2852-2858. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14344.

[40] PLAPPERT M, ANDRYCHOWICZ M, RAY A, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research[J/OL]. CoRR, 2018, abs/1802.09464. http://arxiv.org/abs/1802.09464.

[41] HAARNOJA T, PONG V, ZHOU A, et al. Composable deep reinforcement learning for robotic manipulation[C/OL]//2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018. IEEE, 2018: 6244-6251. https://doi.org/10.1109/ICRA.2018.8460756.

[42] HAARNOJA T, TANG H, ABBEEL P, et al. Reinforcement learning with deep energy-based policies[C/OL]//PRECUP D, TEH Y W. Proceedings of Machine Learning Research: volume 70 Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. PMLR, 2017: 1352-1361. http://proceedings.mlr.press/v70/haarnoja17a.html.

[43] PENG X B, ABBEEL P, LEVINE S, et al. Deepmimic: example-guided deep reinforcement learning of physics-based character skills[J/OL]. ACM Trans. Graph., 2018, 37(4): 143:1-143:14. https://doi.org/10.1145/3197517.3201311.

[44] LEE J, DOSOVITSKIY A, BELLICOSO D, et al. Learning agile and dynamic motor skills for legged robots[J/OL]. Sci. Robotics, 2019, 4(26). https://doi.org/10.1126/scirobotics.aau5872.

[45] NAGABANDI A, CLAVERA I, LIU S, et al. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning[C/OL]//7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. https://openreview.net/forum?id=HyztsoC5Y7.

[46] TIAN S, EBERT F, JAYARAMAN D, et al. Manipulation by feel: Touch-based control with deep predictive models[C/OL]//International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019. IEEE, 2019: 818-824. https://doi.org/10.1109/ICRA.2019.8794219.

[47] AMIRANASHVILI A, DOSOVITSKIY A, KOLTUN V, et al. Motion perception in reinforcement learning with dynamic objects[J/OL]. CoRR, 2019, abs/1901.03162. http://arxiv.org/abs/1901.03162.

[48] LEE M A, ZHU Y, SRINIVASAN K, et al. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks[C/OL]// International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019. IEEE, 2019: 8943-8950. https://doi.org/10.1109/ICRA.2019.8793485.

[49] TAI L, ZHANG J, LIU M, et al. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning[C/OL]//2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018. IEEE, 2018: 1111-1117. https://doi.org/10.1109/ICRA.2018.8460968.

[50] HO J, ERMON S. Generative adversarial imitation learning[C/OL]//LEE D D, SUGIYAMA M, VON LUXBURG U, et al. Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. 2016: 4565-4573. https://proceedings.neurips.cc/paper/2016/hash/cc7e2b878868cbae992d1fb743995d8f-Abstract.html.

[51] TAI L, LIU M. A robot exploration strategy based on q-learning network[C/OL]// IEEE International Conference on Real-time Computing and Robotics, RCAR 2016, Angkor Wat, Cambodia, June 6-10, 2016. IEEE, 2016: 57-62. https://doi.org/10.1109/RCAR.2016.7784001.

[52] TAI L, PAOLO G, LIU M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation[C/OL]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017. IEEE, 2017: 31-36. https://doi.org/10.1109/IROS.2017.8202134.

[53] ZHU Y, MOTTAGHI R, KOLVE E, et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning[C/OL]//2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017. IEEE, 2017: 3357-3364. https://doi.org/10.1109/ICRA.2017.7989381.

[54] KOLVE E, MOTTAGHI R, GORDON D, et al. AI2-THOR: an interactive 3d environment for visual AI[J/OL]. CoRR, 2017, abs/1712.05474. http://arxiv.org/abs/1712.05474.

[55] GRISETTI G, STACHNISS C, BURGARD W. Improved techniques for grid mapping with rao-blackwellized particle filters[J/OL]. IEEE Trans. Robotics, 2007, 23(1): 34-46. https://doi.org/10.1109/TRO.2006.889486.

[56] PAN S J, YANG Q. A survey on transfer learning[J/OL]. IEEE Trans. Knowl. Data Eng., 2010, 22(10): 1345-1359. https://doi.org/10.1109/TKDE.2009.191.

[57] ZHU J, PARK T, ISOLA P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C/OL]//IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. IEEE Computer Society, 2017: 2242-2251. https://doi.org/10.1109/ICCV.2017.244.

[58] GOODFELLOW I J, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial networks[J/OL]. Commun. ACM, 2020, 63(11): 139-144. https://doi.org/10.1145/3422622.

[59] HA D, SCHMIDHUBER J. Recurrent world models facilitate policy evolution [C/OL]//BENGIO S, WALLACH H M, LAROCHELLE H, et al. Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. 2018: 2455-2467. https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

[60] PATHAK D, AGRAWAL P, EFROS A A, et al. Curiosity-driven exploration by self-supervised prediction[C/OL]//PRECUP D, TEH Y W. Proceedings of Machine Learning Research: volume 70 Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. PMLR, 2017: 2778-2787. http://proceedings.mlr.press/v70/pathak17a.html.

[61] GAMRIAN S, GOLDBERG Y. Transfer learning for related reinforcement learning tasks via image-to-image translation[C/OL]//CHAUDHURI K, SALAKHUTDINOV R. Proceedings of Machine Learning Research: volume 97 Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. PMLR, 2019: 2063-2072. http://proceedings.mlr.press/v97/gamrian19a.html.

[62] MIROWSKI P, PASCANU R, VIOLA F, et al. Learning to navigate in complex environments[C/OL]//5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017. https://openreview.net/forum?id=SJMGPrcle.

[63] MIROWSKI P, GRIMES M K, MALINOWSKI M, et al. Learning to navigate in cities without a map[C/OL]//BENGIO S, WALLACH H M, LAROCHELLE H, et al. Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. 2018: 2424-2435. https://proceedings.neurips.cc/paper/2018/hash/e034fb6b66aacc1d48f445ddfb08da98-Abstract.html.

简历与科研成果

基本信息

高可攀，男，汉族，1996 年 9 月出生，江苏省无锡人。

教育背景

2018 年 9 月 — 2021 年 6 月 南京大学计算机科学与技术系 硕士

2014 年 9 月 — 2018 年 6 月 苏州大学计算机科学与技术学院 本科

攻读硕士学位期间完成的学术成果

1. Junyi An, Fengshan Liu, Jian Zhao, Furao Shen, **Kepan Gao**, "IC Neuron: An Efficient Unit to Construct Neural Networks" in *Neural networks, under review*.

攻读硕士学位期间的专利成果

1. 申富饶，**高可攀**，刘小亮，李俊，赵健. "一种融合 UWB 和 LiDAR 的室内定位方法"（202011520518）

攻读硕士学位期间参与的科研课题

1. 国家自然科学基金"基于深度感知增量式联想记忆神经网络的信息融合系统研究"（课题年限 2019.01-2022.12），负责移动机器人定位和导航相关问题的研究。

致　　谢

时光荏苒，白驹过隙，转眼间研究生三年已接近尾声，距离毕业的日子越来越近。回望过去的时间，时光匆匆，一路走来，学到了很多知识，科研能力得到了锻炼，同时也遇到了许多值得感谢的人。在此我要向他们表示深深的感谢，是他们造就了今天的我。

首先需要感谢我的导师申富饶教授。申老师渊博的知识、兢兢业业的态度和严谨的学风给我留下了深刻的印象，申老师在科研上对我们严格要求、悉心指导，教导我们科研要抓住问题的本质。申老师即使再忙也会抽出时间每周与我们单独讨论科研上遇到的问题，研究下一步的规划，在他的指导下，我意识到了科研以及未来漫长的人生中，独立思考的同时也要注意与人交流探讨。同时申老师也在生活上给予我们关心，教导我们为人处世的道理，这是我终生受益的。我也要感谢赵健老师，赵老师在组会上的指导建议给了我巨大的帮助，赵老师细致认真的工作态度值得我学习。

三年的硕士生活中，我还要感谢课题组的其他同学们，各位同学一同科研，互相勉励，常常碰撞出思想的火花，他们的陪伴，让我的硕士生活丰富多彩。感谢他们与我一同经历起起伏伏，在我失落时向我伸出援手。

最后，感谢父母，感谢我的家人，是他们对我无微不至的关心和陪伴，才使我不断进步和成长，能够自由追逐梦想。

学位论文出版授权书

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》（以下简称“章程”），愿意将本人的学位论文提交“中国学术期刊（光盘版）电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版，并同意编入《中国知识资源总库》，在《中国博硕士学位论文评价数据库》中使用和在互联网上传播，同意按“章程”规定享受相关权益。

作者签名：高可攀

_____年_____月_____日

论文题名	基于深度强化学习的移动机器人导航研究				
研究生学号	MG1833020	所在院系	计算机科学与技术系	学位年度	2021
论文级别	☑ 学术学位硕士　☐ 专业学位硕士 ☐ 学术学位博士　☐ 专业学位博士　（请在方框内画勾）				
作者 Email	gaokp@smail.nju.edu.cn				
导师姓名	申富饶				

论文涉密情况：

☑ 不保密

☐ 保密，保密期：_____年_____月_____日　至　_____年_____月_____日

注：请将该授权书填写后装订在学位论文最后一页（南大封面）。