

机器人避障越障的强化学习方法
Reinforcement Learning Methods for Robot Obstacle
Avoidance and Surmounting

学 位 申 请 人： 张俊东
专 业 名 称： 计算机科学与技术
导师姓名及职称： 魏天骐 副教授

答辩委员会主席（签名）：
委员（签名）：

论文原创性声明 & 学位论文使用授权声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

本人完全了解中山大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版；有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆、院系资料室被查阅；有权将学位论文的内容编入有关数据库进行检索；可以采用复印、缩印或其他方法保存学位论文；可以为建立了馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

保密论文保密期满后，适用本声明。

学位论文作者签名：

导师签名：

日期： 年 月 日

日期： 年 月 日

论文题目：机器人避障越障的强化学习方法

专业：计算机科学与技术

硕士生：张俊东

指导教师：魏天骐 副教授

摘要

本论文聚焦于机器人避障与越障任务中的强化学习方法，旨在提升智能体在复杂、动态环境中的适应性与决策能力。随着强化学习在机器人控制、自动驾驶等领域的广泛应用，如何应对动态障碍、复杂地形以及长期决策任务中的挑战成为关键问题。本文从这三方面入手，提出了三种创新性的方法，分别针对动态障碍规避、越障任务中的策略优化以及长期决策任务中的记忆机制，推进了强化学习在机器人自主导航中的应用和理论发展。

首先，针对机器人在动态环境中的障碍规避问题，本文提出了一种结合后见经验回放（HER）与目标障碍相对观测的强化学习算法。该方法通过引入障碍物概念，增强了智能体对复杂环境中动态障碍物的适应能力，能够在稀疏奖励情况下实现高效的避障策略，有效提升了无人驾驶系统和机器人避障任务中的表现。其次，针对越障任务中多峰策略优化的挑战，本文提出了一种基于离散化 Actor-Critic 结构的强化学习模型。通过离散化动作空间，该模型成功避免了连续动作空间模型中可能陷入的单一峰值陷阱，优化了复杂地形中的策略多样性与控制准确性，表现出较高的任务适应性与稳定性。最后，为应对机器人任务中的长期依赖决策问题，本文引入了布尔网络与序列建模方法，提出了一种基于记忆机制的强化学习框架。该框架通过捕捉关键状态转移与记忆结构，提升了智能体在多步推理任务中的推断能力，尤其在复杂情境下的决策效率和准确性上有所提升。

本文的研究表明，所提出的避障算法、离散化 Actor-Critic 模型和基于记忆机制的序列建模方法，均在不同类型的机器人任务中取得了显著的成果，特别是在多样化场景下的智能体适应性与决策能力的提升方面具有重要价值。通过实验验证，所提方法在动态障碍规避、越障任务控制以及长期依赖决策任务中均表现出较好的鲁棒性和有效性，为强化学习在复杂机器人任务中的应用提供了新的思路和技术路径。未来工作将进一步探索如何将这些方法有机整合，构建更为灵活与鲁棒的智能体模型，以应对更加复杂和

多变的实际应用场景，为强化学习技术在实际机器人系统中的应用奠定坚实的理论基础与技术支持。

[关键词] 强化学习；避障；越障；机器人学习

Title: Reinforcement Learning Methods for Robot Obstacle Avoidance and Surmounting

Major: Computer Science

Name: Jundong Zhang

Supervisor: Dr. Tianqi Wei

ABSTRACT

This thesis focuses on reinforcement learning methods for robot obstacle avoidance and terrain surmounting, aiming to enhance the adaptability and decision-making capabilities of agents in complex and dynamic environments. With the widespread application of reinforcement learning in fields such as robotic control and autonomous driving, addressing challenges in dynamic obstacle avoidance, complex terrain surmounting, and long-term decision-making has become a critical issue. This study tackles these three aspects and proposes three innovative methods targeting dynamic obstacle avoidance, strategy optimization for terrain surmounting tasks, and memory mechanisms for long-term decision-making, advancing the application and theoretical development of reinforcement learning in autonomous robot navigation.

First, to address the problem of obstacle avoidance in dynamic environments, this thesis proposes a reinforcement learning algorithm that integrates hindsight experience replay (HER) with relative observations of target obstacles. By introducing the concept of obstacles, the proposed method enhances the agent's ability to adapt to dynamic obstacles in complex environments. It achieves efficient obstacle avoidance strategies under sparse reward conditions, significantly improving the performance of autonomous driving systems and robot obstacle avoidance tasks. Second, for the challenge of optimizing multi-modal strategies in terrain surmounting tasks, this thesis introduces a reinforcement learning model based on a discrete actor-critic structure. By discretizing the action space, the model successfully avoids the single-peak trap common in continuous action space models, optimizing strategy diversity and control accuracy in complex terrains. The model demonstrates high task adaptability and stability. Finally, to address the issue of long-term dependency in decision-making tasks, this thesis incorporates Boolean networks and sequence modeling methods, proposing a reinforcement learning framework based on memory mechanisms. This framework captures key state transitions and

memory structures, improving the agent's reasoning capabilities in multi-step inference tasks, particularly in decision efficiency and accuracy under complex scenarios.

The study demonstrates that the proposed obstacle avoidance algorithm, discrete actor-critic model, and sequence modeling method based on memory mechanisms achieve significant results across various robotic tasks, particularly in enhancing agent adaptability and decision-making capabilities in diverse scenarios. Experimental validation shows that the proposed methods exhibit robust and effective performance in dynamic obstacle avoidance, terrain surmounting control, and long-term dependency decision-making tasks. These findings provide new insights and technical approaches for applying reinforcement learning to complex robotic tasks. Future work will explore how to organically integrate these methods to build more flexible and robust agent models, addressing increasingly complex and dynamic real-world application scenarios. This will lay a solid theoretical foundation and provide technical support for the application of reinforcement learning in practical robotic systems.

[Keywords] Reinforcement Learning; Obstacle Avoidance; Obstacle Surmounting; Robot Learning

目录

摘要	I
ABSTRACT	III
第 1 章 引言	1
1.1 选题背景与意义	1
1.2 国内外研究现状和相关工作	2
1.3 本文的研究内容与主要工作	4
第 2 章 基于强化学习的避障算法	7
2.1 背景	7
2.2 模型与方法	12
2.3 实验	18
2.4 总结与讨论	20
第 3 章 面向机器人步态控制与越障的离散化强化学习	23
3.1 背景	23
3.2 模型与方法	28
3.3 实验	33
3.4 总结和讨论	42
第 4 章 布尔网络与序列建模	45
4.1 背景	45
4.2 模型与方法	46

4.3 实验	49
4.4 总结与讨论	51
第 5 章 结语	55
参考文献	57
攻读硕士学位期间相关的科研成果目录	63
致谢	65

插图

2-1 无障碍物完成目标	10
2-2 有障碍物完成目标	10
2-3 避障仿真环境和模型运行效果	19
2-4 Q 网络损失函数	20
2-5 Q 网络输出均值	20
2-6 局部避障	22
3-1 离散电压会产生持续电流	23
3-2 C51 更新 Z 的操作。改编自 ^[1] 。	26
3-3 算法框架总览	28
3-4 孳生 Critic 网络和离散值分布及其更新方法示意	30
3-5 奶酪陷阱问题	34
3-6 The BipedalWalkerHardcore-v3 任务	37
3-7 BipedalWalkerHardcore-v3 任务中的训练奖励	38
3-8 Walker2d 任务对比：上为 TD3 策略，下为本模型策略	39
3-9 MuJoCo 运动姿态	40
3-10 MuJoCo 训练曲线	41

第1章 引言

1.1 选题背景与意义

深度强化学习（Deep Reinforcement Learning, DRL）目前已经成为机器人控制领域最重要且最新颖的方法论之一。与传统的控制理论相比，强化学习作为现代机器学习的重要分支，更强调智能体在与环境交互中进行学习。这意味着，当将强化学习算法应用于控制系统时，不再需要对系统的状态空间方程进行繁重的人工分析，而只需以数据驱动的方式即可得到性能良好的控制律^[2,3]。不仅如此，强化学习所描述的学习过程与认知科学中的认知过程有高度一致性，它们都强调与环境交互的重要性^[4]。这一点与强调超大数据的监督学习或自监督学习尤其不同。强化学习对于帮助人们更好地理解生物的认知和决策过程具有重要的借鉴意义。例如，研究人员可以利用强化学习框架直观地模拟生物行为的演化过程，观察某些生物学结构或特性对认知与决策过程的影响^[5,6]。

近年来，人工智能技术取得了飞跃式发展，主要得益于大语言模型及其他生成式人工智能所带来的新颖人机交互体验^[7]。与自然语言处理（NLP）和计算机视觉（CV）等炙手可热的领域相比，强化学习领域的热度相对较低。然而，与这些领域的模式识别任务不同，强化学习更加关注智能体的行为决策过程，它依然是构建成熟人工智能系统通向现实物理世界过程中不可或缺的一块拼图，也是推动人工智能能力达到更高水平的一把钥匙^[8]。

机器人避障与越障任务作为强化学习的重要应用之一，其意义不可忽视。在多变的环境中，机器人不仅需要避开障碍物，还需要在复杂地形中找到适当的路径进行越障，这要求机器人能够实时评估周围环境，并作出快速、精确的决策。传统的机器人控制方法往往依赖于精确建模和规则驱动的策略，这在动态、未知的环境中容易失效。而强化学习通过自主探索和实时反馈，能够有效应对这一挑战，为机器人在复杂环境中的自适应行为提供了新的解决方案。因此，研究基于强化学习的机器人避障与越障方法，不仅对机器人导航和自主控制系统的发展具有深远意义，而且推动了机器人在未知环境中智能化、灵活化的应用进程。

尽管强化学习的数学理论已经相对成熟，与现代深度学习结合的过程中仍存在许多不完善之处。比如，为了解决复杂的决策任务，深度强化学习通常会引入两个主要的人工神经网络，分别是 Critic 网络和 Actor 网络，其不同的实现技术会对模型性能产生显著影响^[9,10]。深度强化学习的收敛性是一个重要且值得关注的问题，这在监督学习或自监督学习中通常并不存在。这是因为在强化学习中，策略迭代和策略改进是通过 Critic 网络和 Actor 网络之间的信息交互完成的，与端到端学习模式有着本质的区别。因此，深度强化学习对超参数选择往往非常敏感，这反映了目前对其理论与技术的理解尚有不足之处。

在现有技术条件下，开发更多强化学习算法以应对实际问题具有重要意义。例如，针对组合离散动作空间（又称大离散动作空间）的研究，可以有效扩展强化学习在更一般应用场景中的能力，改善问题解决范式^[11,12]。

1.2 国内外研究现状和相关工作

机器人避障（Obstacle Avoidance）是机器人自主导航领域的核心问题之一，其目标是在未知或动态环境中，通过实时感知与路径规划，确保机器人能够安全、稳定地避开障碍物。传统的避障方法多基于几何规则或启发式规划，例如 Dijkstra 算法、A* 算法和人工势场法^[13,14]。这些方法对环境建模和传感器精度要求较高，难以在复杂和动态环境中保持稳定性能。

机器人越障（Terrain Surmounting）主要聚焦于复杂地形下的运动控制与导航问题，通常涉及非结构化环境中的路径规划和决策优化。越障任务的技术难点在于，机器人需要在不确定的地形中保持平衡，同时进行精确的运动控制。传统的越障方法多依赖于运动学与动力学建模，通过优化运动轨迹或增加额外传感器来应对复杂地形^[15]。

尽管传统方法在机器人避障与越障中取得了一定成效，但这些方法往往依赖于精确的环境建模和规则驱动的控制策略，对未知或动态场景的适应性较差。相比之下，强化学习通过智能体与环境的交互学习，能够在无模型或弱建模的条件下直接优化策略，显著提升了机器人在复杂任务中的适应性与鲁棒性^[3]。此外，强化学习不仅适用于连续的运动控制问题，还能够在离散化动作空间中有效处理多样化策略分布的问题，使其在避障与越障等复杂任务中具有广泛的应用潜力。

深度确定性策略梯度 (Deep Deterministic Policy Gradient, DDPG)^[16] 是一种广泛应用于连续动作空间的深度强化学习算法。DDPG 属于离线策略算法，因此通常与重放缓冲区配合使用，缓冲区存储从环境中收集的每个状态转移，包括当前状态、动作、下一个状态、奖励以及是否回合结束。当缓冲区满时，旧数据会被清除，确保算法能及时学习新的经验。

DDPG 由两个核心部分组成：Actor 网络和 Critic 网络。Actor 网络 $\pi_\theta : S \rightarrow A$ 的目标是根据当前状态选择最优动作，从而最大化 Critic 网络 Q_ϕ 输出的动作价值。Critic 网络的任务是通过最小化 $L_\phi = \mathbb{E}(\|y_t - Q_\phi(s_t, a_t)\|_2^2)$ 来近似动作价值函数 Q^π ，其中 $y_t = r_t + \gamma Q_\phi(s_{t+1}, \pi_\theta(s_{t+1}))$ 。需要注意的是，由于半梯度下降的特性，Critic 的更新与 y_t 之间没有直接的梯度关系，这使得训练过程中容易出现不稳定性，尤其是在高方差奖励的环境中。

为了解决 DDPG 中的值高估问题，双重延迟深度确定性策略梯度 (Twin Delayed DDPG, TD3)^[17] 提出了几个关键的改进。首先，TD3 使用了双 Q 网络来通过截断机制减少高估误差；其次，它通过平滑目标网络和减少策略更新频率来增强算法的稳定性。这些改进在高方差奖励环境中尤其有效，显著提升了训练的鲁棒性。

在稀疏奖励问题上，后见经验回放 (Hindsight Experience Replay, HER)^[18] 为深度强化学习提供了新的思路。HER 通过在任务失败时重新定义目标，使得即使失败的经验也能为学习提供有效信息。然而，HER 的应用也有其局限性。首先，环境必须支持动态目标切换；其次，HER 假设目标是环境的唯一影响因素，但实际上，障碍物、地形等环境设置对任务的完成至关重要，这些因素并未在 HER 框架中充分考虑。

这些方法的演进使得主流深度强化学习算法呈现出递进关系：DDPG 作为基础方法解决了连续控制问题，TD3 和 SAC 则在其基础上进一步改进，提升了训练的稳定性和探索能力。特别是 SAC^[9] 通过引入熵正则化和随机 Actor 网络，显著增强了策略的探索能力。此外，TQC^[19] 基于 QR-DQN^[20] 引入了分布式值表示，并通过截断混合机制有效减轻了高估问题。DDPG、TD3、SAC、TQC 以及 PPO 在如 MuJoCo 任务^[21] 中的表现均优异，且这些算法的实现可通过 Stable Baselines 3^[22] 等工具库获得，后续章节将比较它们在实验中的表现。

与传统方法不同，值分布强化学习模型专注于建模奖励分布，而非仅仅估计期望值。

C51^[1] 使用离散的价值分布构建 Critic 网络，而 QR-DQN^[20] 和 IQN^[23] 则通过分位数回归方法进一步精细化奖励回报的分布。这些方法不仅在理论上通过 Wasserstein 度量验证了收敛性^[24]，而且在 Atari 任务中展现了优越的性能。D4PG^[25] 结合了 C51 和 Actor 网络，在深度强化学习中开辟了新的研究方向。

更多的相关工作包括：SAC-Discrete^[26] 将 SAC 的范围扩展到离散动作空间，从而增强了模型利用动作熵进行探索的能力。文献^[27] 探讨了一种基于 MPO 算法的离散与连续动作变量统一控制的可行方法。文献^[28] 将多维动作变量分解为一系列离散变量的决策过程。文献^[29] 倡导对连续动作空间进行离散化，这可以提升诸如 PPO 等在线算法的性能。文献^[30] 强调在离线强化学习中离散化动作空间的好处，并考察潜在解决方案。文献^[31] 认为，与均方误差损失函数相比，交叉熵损失函数在强化学习中训练 Critic 网络时更有效，特别是对于参数数量较大的模型，如 Transformers。

综合来看，深度强化学习算法的研究不断朝着更高效、稳定的方向发展，尤其是在处理复杂环境（如稀疏奖励、高方差奖励和高维动作空间）方面取得了显著进展。然而，当前的算法仍面临在复杂地形和动态环境下的挑战，这也为后续的研究提供了丰富的改进空间。

1.3 本文的研究内容与主要工作

强化学习作为机器学习的重要分支，旨在通过智能体与环境的交互学习最优决策策略。其研究框架通常包括问题建模、算法设计与优化、实验验证及应用推广等环节。在问题建模阶段，通过构建马尔可夫决策过程（MDP）形式化问题定义，明确状态空间、动作空间、转移概率和奖励函数等核心要素。在算法设计中，基于价值函数、策略优化或两者结合的范式开发高效学习算法。实验验证环节则利用仿真环境或实际场景评估算法性能，从多个维度验证其收敛性、鲁棒性和泛化能力。

研究强化学习的可行性主要体现在以下几个方面：首先，强化学习拥有严格的理论基础，具备解析复杂动态系统的能力；其次，随着计算能力和深度学习技术的发展，其应用范围逐步拓展，适用于机器人控制、游戏智能体设计及资源调度等领域。研究的可靠性则通过算法在多样化任务中的稳定表现、对噪声干扰的抵抗能力以及对真实世界问题的有效适配来保证。此外，强化学习研究的进一步发展依赖于算法效率的提升和对复

杂环境适应性的增强，为解决现实问题提供创新方法。

在上述研究框架和技术路线的基础上，本文展开了以下几项工作。

第 2 章介绍强化学习在无人驾驶中实现避障的应用，提出了一种兼顾目标与障碍物的训练方法。在 HER+DDPG 的框架下，引入了障碍物的概念。本节所介绍的策略探索方法既兼顾了原框架下对于稀疏奖励的经验构造，还允许模型就环境内的障碍物搜索多样化解决方案。但由于稀疏的高奖励信号和高惩罚信号共存，为模型的收敛带来了新的困难，本节所介绍的奖励截断的方法可以有效稳定训练的过程。

第 3 章聚焦于机器人越障任务，深入探讨了离散强化学习模型在应对复杂地形中的优势。本章首先分析了主流强化学习模型在越障任务中的主要局限性，尤其是在处理复杂地形特性时的不足。针对这些问题，提出了一种基于离散化动作空间的强化学习算法，用于优化机器人在复杂地形中的越障策略。具体而言，在奶酪陷阱任务中，以 SAC 为代表的主流模型未能成功完成任务，揭示了多峰动作分布建模的必要性。本章实验表明，通过离散化动作空间模拟多峰动作分布，无需对不同地形单独设计训练课程，即可显著提升模型在复杂地形越障任务中的表现。此外，本章提出的离散价值分布方法相比第 2 章中采用的连续动作模型（如 DDPG）更具鲁棒性，尤其是在奖励函数处理方面表现突出。为进一步提升算法性能，引入了孪生 Critic 网络，减少了价值高估的问题。同时，设计了适配离散化动作空间的 Actor 模块，并配备了定时遍历探索和失败驱动探索等机制，使智能体在复杂地形中的策略优化更加高效。在广泛认可的 MuJoCo 任务集中进行的测试表明，该模型在越障任务中的表现与此前主流模型相比显著提升，验证了所提出方法的有效性和普适性。

第 4 章探索和展望部分可观察马尔可夫模型（POMDP）未来可能的发展方向，着重考察记忆力形成的机理，以增强强化学习模型在复杂上下文场景中的理解能力。本节尝试在深度学习和概率论的基础框架之外，开发一种新的机器学习算子，该算子借鉴了表示论与布尔网络等领域的相关思想，通过构造从序列元素到布尔矩阵的同态映射，以表征非线性系统的动态特性。实验结果表明，该模型在三个基本记忆任务中的状态转移学习性能相较于 Transformer 模型有一定提升。

这种记忆结构与部分可观察场景中的任务规划密切相关，尤其在机器人越障和避障任务中具有重要潜力。在实际应用中，机器人经常需要根据不完全的环境观测信息进行

决策，尤其是在复杂地形下的越障过程中，路径选择和策略优化常涉及多步推断与长程依赖关系。通过增强强化学习模型的记忆能力与上下文理解能力，有望在动态避障和多目标越障任务中实现更高效的任务规划和策略生成。此外，该算子对非线性系统的表征能力，也为未来在动态环境中的任务自适应优化奠定了理论基础，进一步推动机器人在部分可观察任务中的性能提升。

本论文从应用实践到方法优化，再到理论探索，形成了层层递进的逻辑结构。第 2 章以无人驾驶避障为例，提出了兼顾稀疏奖励和障碍物复杂性的强化学习方法，为解决高维动态环境中的策略问题奠定基础；第 3 章进一步扩展至机器人复杂地形控制，利用离散化动作空间提升模型的适应性和鲁棒性，面向不同的任务场景进行优化；第 4 章则面向未来，探索可观察马尔可夫模型在记忆机制建模中的潜力，为强化学习的长期依赖关系和复杂上下文理解提供理论支持。各章节紧密关联，共同探讨了强化学习从具体任务到一般方法的系统性提升。

第 2 章 基于强化学习的避障算法

2.1 背景

避障任务广泛应用于自动驾驶、机器人导航以及无人机飞行等领域，是确保智能系统安全性和实用性的关键环节。在自动驾驶中，车辆需要实时感知周围环境，并规避可能的碰撞风险，以保障行车安全。在机器人导航中，避障能力直接决定了机器人在动态环境中自主移动的效率与可靠性，特别是在工业自动化、物流配送和家庭服务等场景中，无障碍运行至关重要。类似地，在无人机飞行中，避障技术不仅能够减少与障碍物碰撞的可能性，还可以帮助无人机在复杂环境中规划最优路径。这些场景对避障策略提出了高实时性与高鲁棒性的要求，传统基于规则的方法难以适应高度复杂和动态的环境，而强化学习则为解决这些问题提供了一种数据驱动的有效手段。

无人驾驶车辆避障是自动驾驶技术的重要组成部分，已有多种经典方法被提出。基于规则的方法依赖于预定义的启发式规则，通过传感器输入实现实时避障，如动态窗口法（Dynamic Window Approach, DWA）^[32]，适用于结构化环境但难以处理动态场景。基于路径规划的方法通过全局与局部规划相结合生成最优路径，其中 A* 算法、快速扩展随机树（RRT）等被广泛应用^[33]。模型预测控制（Model Predictive Control, MPC）方法利用优化技术预测系统未来行为，可有效处理运动学约束，但实时计算需求较高^[34]。

在这些方法中，通常会选择一些基于可靠动力学模型的方法，然后利用控制理论中的一些结论来构建控制律。如何在底层控制器中考虑障碍物的影响仍然是一个有挑战的问题。实际上，可靠的动力学模型很难获取，这依赖于工程师的分析能力。相反，大多数强化学习算法是无模型的，这意味着强化学习方法可能更适应不同类型的移动机器人。

强化学习（RL）是一个关于智能体在与环境互动中学习的领域。强化学习与深度学习的结合催生了许多知名的基本算法，例如深度确定性策略梯度（DDPG）^[16]。这些算法被用于解决许多有趣的问题，例如控制机器人和玩游戏。然而，由于许多因素，深度强化学习的应用仍然有限。

首先，设计一个高质量的奖励函数在实践中是很困难的，而这是生成最优策略所必需的。设计奖励函数最简单的方法之一是，在检测到智能体达成目标时给予高奖励，而在智能体做了禁止的事情时给予严厉的惩罚。虽然这种方法有其优势，但它也导致奖励的分布存在巨大方差，这可能使模型难以收敛。

其次，探索问题也是强化学习中的一个关键问题。如果智能体的初始策略（通常是在行动空间中的随机策略）仅有极小的概率实现目标，强化学习算法将退化为一种低效的随机搜索方法。 ϵ -贪婪策略和好奇机制^[35] 旨在解决一些探索问题，但在处理稀疏奖励时，它们仍然需要花费大量时间在观察空间中进行搜索。

再者，强化学习中的马尔可夫决策过程（MDP）框架可能限制了其交互模式。后见经验回放（HER）^[18] 建议，除了 MDP 之外，可以将目标向量视为观察的一部分，这样即使智能体未能实现原始目标，也可以将原始目标转换为智能体的当前位置，并将转换后的轨迹存储为额外的学习经验。同时，MDP 通常限制智能体在相对固定的场景中移动，这意味着一旦环境发生变化，智能体可能无法适应，并且尽管不同情境之间极为相关，智能体可能无法转移其技能。

后见经验回放（Hindsight Experience Replay, HER）^[18] 是一种有效解决稀疏奖励问题的方法。然而，为了应用 HER，环境需要满足一些额外的要求，即它应该支持假设已经收集了当前策略的轨迹 $tj = \{(s_t|g, a_t|g, s_{t+1}|g, r_t|g, d_t|g) | t = 0, 1, \dots\}$ （注： $s_t|g$ 表示在目标 g 前提下的状态观察变量 s_t 的值，不同的目标会有不同的值，以此类推）。当发现策略根本无法帮助实现期望的目标 g ，并失败了并达到了一个不同的位置 g' 时，可以构建一个额外的轨迹 $tj' = \{(s_t|g', a_t|g, s_{t+1}|g', r_t|g', d_t|g') | t = 0, 1, \dots\}$ ，并将其作为额外经验添加到重放缓冲区中以供学习。需要提醒的是，切换目标有时需要更多额外的计算来转换观察和奖励，因为在某些情况下，目标可能不直接包含在观察中。另外，在 HER 框架下没有考虑除目标以外的其他环境设置（如障碍物、地形、禁区等），有必要在此基础上有所扩展。

最近，一些研究尝试在强化学习中探索超越 MDP 的可能性。可配置马尔可夫决策过程（Conf-MDPs）^[36] 建议，不仅可以从环境返回的奖励中学习，还可以使环境可配置。可配置环境是指在开始新一轮之前，可以配置其中的一些因素，而这些因素通常对智能体的策略有很大的影响。

在本文中，希望通过相关理论来解决避碰问题。本文将尝试将避碰问题描述为一个可配置环境。然后，将介绍在训练过程中遇到的一些问题，包括稀疏奖励、数值爆炸、探索和遗忘。还将讨论一些缓解这些现象的技巧。本文的贡献包括：

- 尝试训练一个在可配置环境中工作的单一智能体。
- 针对高方差奖励的归一化方法。
- 附加经验构建和人类设计的探索策略框架。
- 应用优先级经验回放 (PER)^[37]，使智能体在训练过程中关注重要案例。
- 一个同时考虑目标和障碍的底层控制器。

2.1.1 任务环境

在本节中，将详细讨论环境。CarRacing-v2^① 是一个由 OpenAI Gym 提供的有趣演示环境，旨在进行强化学习研究。在该环境中，汽车应尽快沿着赛道行驶并访问更多的瓦片。当汽车到达一个瓦片时，它可以获得高奖励。同时，随着时间的推移，汽车会受到小惩罚，以激励智能体尽快完成任务。

CarRacing-v2 中的奖励实际上是密集的，因为赛道上有很多瓦片。因此，可以发现许多基本的强化学习算法在这个任务中表现得非常好。接下来，将 CarRacing-v2 转变为一个更有趣的版本，使用稀疏奖励，并且可能不再要求智能体沿着赛道行驶，而是执行其他任务。车辆的动力学建模可概括为：具有四个轮子用于移动和两个前轮用于转向，类似于在生活中看到的普通汽车。实现方法可以参考该教程^②。

2.1.2 任务观测数据

与 CarRacing-v2 不同，在这里使用的是向量观察，而不是像素观察。该向量的含义如下：(1) 整辆车的速度；(2) 左前轮的轮子角速度；(3) 右前轮的轮子角速度；(4) 左后轮的轮子角速度；(5) 右后轮的轮子角速度；(6) 前轮的转向角；(7) 整辆车的角速度；

基础向量观察是由这 7 个值构成的。它是通过 Box2d 物理模拟提供的，模拟了真实汽车中的传感器，表达了汽车的运动状态。随后，为了构建不同的强化学习任务，将向

^① https://www.gymlibrary.dev/environments/box2d/car_racing/

^② <http://www.iforce2d.net/b2dtut/top-down-car>



图 2-1 无障碍物完成目标



图 2-2 有障碍物完成目标

向量观察中添加一些其他值。

2.1.3 任务动作空间

汽车的动作空间是连续的，仅包含 3 个值，包括：

- 目标转向角。汽车会尽快将前轮转向该值。考虑到惯性的存在，转向角不能立即改变。该值范围在 -1 到 1 之间。
- 油门。实际上，CarRacing-v2 只允许汽车向前行驶，油门值只能为正。但为了使事情更有趣，允许油门值为负，这意味着汽车在油门为负时向后行驶，在为正时向前行驶。该值范围在 -1 到 1 之间。
- 刹车。刹车减慢汽车的速度。其值范围在 0 到 1 之间，分别表示不刹车和重刹车。

2.1.4 无障碍任务

受到了 Python Robotics^[38] 提供的一个演示“移动到目标位置”的启发，最终将其与 CarRacing-v2 结合起来。为智能体设置了一个任务，要求它尽快控制汽车到达地面的目标位置和目标角度。目标在每一轮中随机生成。希望智能体能够学会控制汽车到达生成的任意目标。一旦汽车达成目标，环境将返回高达 500 的奖励并结束这一轮，否则将对每个时间步返回 -1 的时间惩罚。此外，为了鼓励智能体向前行驶，还设置了当汽车向后行驶或刹车时的惩罚为 -0.3。

需要在观察中添加一个目标向量。参考 Python Robotics 中的演示，选择将目标的坐

标相对于汽车作为目标向量，而不是绝对坐标。这意味着在 HER 中切换目标时，需要进行一些计算。

在汽车上绑定一个极轴，定义向左转为正角度，向右转为负角度，且该轴的方向与汽车的前进方向相同。然后，可以得到目标的极坐标 $s_g = (\rho_g, \alpha_g, \beta)$ ，其中 (ρ_g, α_g) 是目标位置的极坐标， β 是相对于极轴的目标角度。但这还不算结束，如果目标位置在汽车后方，即 $\alpha_g \notin (-\pi/2, \pi/2)$ ，那么需要进行一些额外的处理。

$$\begin{aligned}\rho'_g &= -\rho_g \\ \alpha'_g &= \text{sign}(\alpha_g)(\pi - |\alpha_g|) \\ s_g &= (\rho'_g, \alpha'_g, \beta)\end{aligned}\tag{2-1}$$

“移动到目标位置”任务的解决方案并不复杂，可以直接使用 HER + DDPG，这样智能体就能快速学习。因此，不会在这方面花费太多篇幅，而是关注包含障碍物的情况。

2.1.5 有障碍任务

可以给智能体增加一些挑战。在地面上放置一个障碍物，当汽车碰到障碍物时，环境会结束这一轮并返回高达 -500 的惩罚。障碍物的形状是一个简单的圆，且只有一个障碍物。程序会确保障碍物与目标或汽车不重叠，如果发生重叠，则将其移动到一个足够接近的位置，保持一个小间隙。这个规则确保智能体至少有一个解决方案，同时给智能体带来碰撞的风险。还应该将障碍物观察向量添加到整个观察向量中。该向量表示相对于汽车的位置和圆的大小。这部分并不是特别重要，有些琐碎。为了与没有障碍物的情况兼容，需要额外定义，当没有障碍物时，障碍物观察为 $s_o = (0, 0, 0, 0)$ 。设 (ρ_o, α_o) 为障碍物的相对极坐标， r_o 为障碍物的半径。然后定义

$$\begin{aligned}\theta &= \arcsin\left(\frac{r_o}{\rho_o}\right) \\ \rho'_o &= \rho_o - r_o \\ s_o &= (1, \rho'_o, \alpha_o, \theta)\end{aligned}\tag{2-2}$$

再者, 如果 $\alpha_o \notin (-\pi/2, \pi/2)$, 那么

$$\begin{aligned}\rho''_o &= -\rho'_o \\ \alpha'_o &= \text{sign}(\alpha_o)(\pi - |\alpha_o|) \\ s_o &= (1, \rho''_o, \alpha'_o, \theta)\end{aligned}\tag{2-3}$$

2.1.6 环境的可配置性

与其他 RL 环境相比, 这样的环境具有一些特殊之处。首先, 在开始新一轮时, 需要向环境输入一些配置参数, 比如目标和障碍物的姿态。此外, 当智能体在不同配置下执行任务时, 观察中的某些因素与配置相关, 这些因素不受智能体任何动作的影响, 并且可能对策略产生很大影响。在例子中, 目标和障碍物之间的相对位置无论智能体将汽车控制到哪里都不会改变。对于这种情况, 可以说环境发生了一些不可抗力的变化, 环境是可配置的。

假设 ζ 是一个合法的配置参数向量, 它能够唯一地确定环境中的一种情况, 那么称 ζ 为 **案例向量**。由 ζ 启动的回合集合称为可配置环境中 ζ 的 **案例**。对于之前讨论的环境, 要启动一轮, 需要确定目标的绝对位置和角度 $g = (x_g, y_g, \delta)$, 以及障碍物的绝对位置和大小 $o = (x_o, y_o, r_o)$ 。那么该环境的案例向量 ζ 为 $\zeta = (g, o)$ 。

2.2 模型与方法

对于这一个基础的经典的控制问题, 人们提出过非常多方法和思路。然而对于这样的问题结合具体运动学模型来求解精确解可能是困难的, 结合最优化理论来对机器人进行运动规划恐怕也是不切实际的, 因为这样的问题往往不具备凸性, 以至于不得不做一些简化的处理, 否则无法保证计算的高效性。近年来, 强化学习在新一轮 AI 浪潮中也得到了发展。在这篇工作中, 提议可以尝试深度强化学习来解决这一问题。参考的工作有:

- 1) DDPG^[16] 连续动作空间基本深度强化学习算法。
- 2) TD3^[17] 利用孪生网络来减少误差。
- 3) Hindsight Experience Replay (HER)^[18] 解决基于目标的强化学习任务的基本思路,

通过数据增强构建可学习的样本，从而突破稀疏奖励带来的限制。

- 4) Prioritized Experience Replay (PER)^[37] 根据拟合误差来构建样本的优先级，帮助 Critic 网络能够更好地拟合样本，对于某些重要的小概率样本可以增强它们在训练中的作用。

2.2.1 Q 值的截断归一化

然而，直接把前人的模型拿过来恐怕还不能直接解决这个问题，必须再做一些额外的工作。稀疏奖励对于模型的收敛性提出了较大的挑战，特别是同时存在高额奖励和高额惩罚。正如前文所示，将奖励函数设计得尽可能简单。如果达到目标，则得 500 分；如果发生碰撞，则得 -500 分；否则每个时间步得 -1 分作为时间惩罚。这样一个具有巨大方差的稀疏奖励函数可能会导致 Q 网络不稳定。即使已经在 TD3 中使用了一些技巧，值爆炸仍可能发生。原因在于巨大的奖励可能导致巨大的梯度，从而损失大量精度。

如果将需要被拟合的 Q 值记为 y_t ，一般而言有， $y_t = r_t + \gamma \min_{i=1,2} Q'(s_{t+1}, \pi'(s_{t+1}))$ 。希望对 y_t 进行一些处理来增加模型的鲁棒性。幸运的是，解决这个问题的技巧相当简单。只需要稍微调整 Q 网络的损失。这个技巧的思想是避免 Q 网络在单次更新迭代中发生过于剧烈的变化，即使这可能导致更多的训练。在实践中，发现这个简单的技巧可以使训练更加稳定和可靠。具体而言，做了以下这些步骤：

- 1) 令 $y'_t = \max(y_t, \sum_{i=t}^{\infty} \gamma^{i-t} r_i)$ ，认为使用 y'_t 可以有效避免 agent 过于悲观，可以加速模型的训练。
- 2) 令 $y''_t = Q(s_t, a_t) + [y'_t - Q(s_t, a_t)]_{-\epsilon}^{+\epsilon}$ ，认为这样的 y''_t 可以将 MSE loss 控制在一个合理的范围 $[-\epsilon, +\epsilon]$ 之中，这样做可以有效增强模型在训练过程的鲁棒性，避免模型坍塌或者不收敛。在别的一些强化学习的工作中，通常称这样的处理为”reward clip”。
- 3) 最后的损失函数的梯度当然就是 $\nabla_{\theta} L_{critic} = \nabla_Q (y''_t - Q(s_t, a_t))^2 \nabla_{\theta} Q(s_t, a_t)$ ，如果以 θ 来表示 Q 网络参数的话。

2.2.2 动作的记录与回放

建议可以记录智能体在一轮中的行为动作。然后，可以在其他案例的回合中展示相同动作序列。这可能有助于为智能体生成一些额外的有意义的经验进行学习。在日常生活中，人们在学习新事物时可以比较不同情况下的相同行为，这正是关键所在。然而，为了节省重放缓冲区的内存并提高训练效率，如果决定向重放缓冲区添加一些额外数据，最好要更加谨慎。必须选择真正有价值的数据，并过滤掉不必要的数据。

假设在一个案例 $\zeta = (g, o)$ 中模型生成了一个轨迹，其中 g 和 o 分别是目标和障碍物的配置，这意味着

$$\begin{aligned}\zeta &= (g, o) \\ tj &= \{(s_t|\zeta, a_t|\zeta, s_{t+1}|\zeta, r_t|\zeta, d_t|\zeta) \mid t = 0, 1, \dots\}\end{aligned}\tag{2-4}$$

智能体可能根本没有达到目标，甚至撞上了障碍物，但是这也没关系，因为智能体正在学习，只是希望智能体能够从失败中吸取教训。假设发现智能体在碰撞之前达到了一个位置 g' ，那么可以将轨迹切换到案例 $\zeta' = (g', o)$ ，并将其添加到重放缓冲区，就像 HER 所做的那样。

$$\begin{aligned}\zeta' &= (g', o) \\ tj' &= \{(s_t|\zeta', a_t|\zeta, s_{t+1}|\zeta', r_t|\zeta', d_t|\zeta') \mid t = 0, 1, \dots\}\end{aligned}\tag{2-5}$$

同样，也可以对障碍物做同样的处理。可以将轨迹切换到想要的任意障碍物，这意味着如果 o' 是另一个障碍物，那么

$$\begin{aligned}\zeta'' &= (g', o') \\ tj'' &= \{(s_t|\zeta'', a_t|\zeta, s_{t+1}|\zeta'', r_t|\zeta'', d_t|\zeta'') \mid t = 0, 1, \dots\}\end{aligned}\tag{2-6}$$

然而，正如之前强调的，需要谨慎保留数据。以下是一些参考原则：对于智能体自己生成的轨迹，应无条件接受。对于构造的额外轨迹，只接受成功的轨迹，这意味着如果轨迹撞上障碍物或未达到目标时应将其丢弃。此外，应保持真实轨迹与构造轨迹之间的合理比例，这意味着如果真实轨迹的比例过低，即使构造的轨迹是成功的，也需要丢弃一些构造轨迹。

在这里介绍的技巧在训练初期可以提供很大帮助。它帮助智能体更快地理解问题。如果去掉这个技巧，使用纯基础的深度强化学习算法，模型根本无法收敛，因为随机策略达到目标的概率极低。然而，建议在训练的后期减少构造轨迹的比例，并且需要更多的技巧来让智能体获得更好的表现。

Hindsight Experience Replay (HER) 依然是强化学习领域应对稀疏奖励的最主要的方法。然而只有 HER 可能还不足以解决这个问题。这是因为 HER 中主要讨论的是 goal-based 的强化学习的问题，而任务中除了目标之外还有障碍物的存在，因此不得不在 HER 的基础上进行一些扩展。工作包括

- 1) 首先只考虑没有障碍物的情况，这时，可以直接应用 HER 来对模型进行训练，将这个不考虑障碍物的模型称作 agent1，agent1 是非常容易训练和收敛的。
- 2) 把考虑了障碍物的模型称作 agent2，相比于 agent1，agent2 的训练困难很多，agent2 也可以使用 HER，当 agent2 从 HER 的机制中学到的东西可能是有限的，除此以外，还需要提供切实有效的探索机制

2.2.3 子目标探索策略

HER 已经讨论了智能体如何从失败中构建成功经验，但还是需要智能体自己发现一些新技能。数据是机器学习的生命。在深度强化学习中，通常不使用数据库中的数据，而是在与环境互动的过程中收集一些数据。因此，值得花一些篇幅来讨论探索问题，因为它直接决定了用来进行训练的数据质量。需要强调的是，在这里介绍的技巧是专门针对碰撞避免问题设计的，基于一些人类的先验知识，可能不适用于其他问题。

一些关于层次强化学习 (HRL) 的工作，例如^[39]，中的一个常见的概念叫做“子目标”。如果智能体想要完成一些复杂的任务，最好将其分解为几个更简单的子目标。这些研究中的一些尝试了不同的网络结构来验证这一观点，这很有启发性。然而，在这里不想讨论神经网络的结构，而是子目标与探索的结合。

假设在案例 $\zeta = (g, o)$ 中收集了智能体的一个轨迹，不幸的是，它发生了碰撞。然后需要问第一个问题，如果让智能体忽略障碍物，会发生什么？用 ϵ 表示没有障碍物，然后可以在案例 $\zeta' = (g, \epsilon)$ 中尝试使智能体收集一个新的轨迹 tj' 。接着，记录 tj' 中的动作序列，并在案例 $\zeta = (g, o)$ 中展示它，看看会发生什么。

$$\begin{aligned}
 \zeta' &= (g, \epsilon) \\
 tj' &= \{(s_t|\zeta', a_t|\zeta', s_{t+1}|\zeta', r_t|\zeta', d_t|\zeta') \mid t = 0, 1, \dots\} \\
 \zeta &= (g, o) \\
 tj &= \{(s_t|\zeta, a_t|\zeta, s_{t+1}|\zeta, r_t|\zeta, d_t|\zeta) \mid t = 0, 1, \dots\}
 \end{aligned} \tag{2-7}$$

如果幸运的话，它达到了目标，那么这次探索就结束了，只需要将 tj 添加到重放缓冲区。但如果还没有达到目标，可以问第二个问题。看起来障碍物阻碍了通往目标的路径，如果让智能体先去别的地方，会发生什么？可以首先生成一个随机的子目标 g' ，并指挥智能体一开始朝这个子目标移动，然后再让智能体继续实现它的原始目标。操作大致如下：

$$\begin{aligned}
 \zeta' &= (g', \epsilon) \\
 tj' &= \{(s_t|\zeta', a_t|\zeta', s_{t+1}|\zeta', r_t|\zeta', d_t|\zeta') \mid t = 0, 1, \dots, n\}
 \end{aligned} \tag{2-8}$$

可以选择一个你喜欢的值作为 n ，例如在 30 到 80 之间的随机值。记录从 0 到 n 时间步的动作序列，并在案例 $\zeta = (g, o)$ 中展示它。然而，经过 n 时间步后，剧集很可能还没有结束，让智能体继续并完成它。

$$\begin{aligned}
 \zeta &= (g, o) \\
 tj^1 &= \{(s_t|\zeta, a_t|\zeta, s_{t+1}|\zeta, r_t|\zeta, d_t|\zeta) \mid t = 0, 1, \dots, n\} \\
 tj^2 &= \{(s_t|\zeta, a_t|\zeta, s_{t+1}|\zeta, r_t|\zeta, d_t|\zeta) \mid t = n + 1, n + 2, \dots\} \\
 tj &= tj^1 \cup tj^2
 \end{aligned} \tag{2-9}$$

如果轨迹成功，就结束这次探索；否则，可以尝试更多的子目标，或者放弃这个案例。在实践中，这个技巧能够帮助发现多种实现目标的方法，对智能体解决相对困难的案例非常有帮助。

对于一个具体的随机生成的目标（goal）和障碍物（obstacle）而言，提出如下的探索机制，它可以帮助模型跳出局限区域，发现可行方案并不断优化

- 1) 如果 agent2 可以控制小车直接到达目标并没有发生碰撞，则直接将这个 episode 的轨迹数据存到 Replay Buffer 中即可

- 2) 如果 agent2 无法到达目标，或者发生了碰撞，那么尝试直接使用 agent1 来从初始位置驶向目标，如果 agent1 的行为是可行的话，那么把 agent1 所生成的轨迹数据作为示范数据存到 Replay Buffer
- 3) 如果 agent2 和 agent1 都无法直接到达目标的话，此时生成一个随机子目标 g_{sub} ，然后尝试使用 agent1 驾驶到 g_{sub} ，然后过了一段时间后，再尝试切换回 agent2 驾驶向原来的目标 g 。这个过程由 agent1 和 agent2 共同协作来完成。这个过程可能会重复有限次，直至能够探索出切实可行的方案。

2.2.4 优先级回放机制

讨论了很多关于数据收集的主题，但仍然存在一些问题。在实验中，智能体似乎出现了遗忘现象。尽管某些案例出现了多次，智能体仍然无法在再次遇到时解决它们，因为这些案例的频率明显低于其他案例。这个现象反应一些低频案例可能更为重要。

通过建立一个经验池，可以将一组经验添加到池中或从池中移除，池的容量是固定的。池中的每组经验都有一个累计得分，用于衡量智能体在该经验组中的掌握程度。每次选择训练智能体的数据都应该是从池中随机抽取。当一组经验进入池时，它将获得一个固定的正初始得分。当智能体完成一个案例的训练回合时，将该回合的拟合误差到其累计得分中。在整个过程中，要确保所有的累计得分都是非负的。当池满了并且尝试添加一组新经验时，池内将找到累计得分最高的案例并将其删除，然后接受新案例。这就是优先级回放机制。

通过优先级回放机制，智能体失败的案例会获得较低的得分，并在池中停留更长时间，因此智能体会更频繁地尝试这些案例。实际上，优先级回放机制改变了案例的经验概率分布。较大的池可以确保案例的概率分布平滑变化，但可能会降低其对遗忘现象的防范效果。相反，较小的池可能导致概率分布的剧烈变化，从而使模型无法收敛。

”reward clip”对于这个任务的收敛性起到了非常关键的作用，配合 Prioritized Experience Replay (PER)^[37]一起使用，可以起到更好的效果。实现 Prioritized Replay Buffer 的方式可能与原论文有所不同，具体而言

- 1) 记录 Replay Buffer 中的每个样本对应的 $|y_t - Q(s_t, a_t)|$ ，作为该样本的 Priority，也就是优先级

- 2) 当 Replay Buffer 没有满时，直接将当前的 transition 元组加到里面去
- 3) 当 Replay Buffer 满了，首先从 Replay Buffer 中抽取 128 个样本，并找出里面优先级最低的样本，并将新的样本代替掉这个优先级最低的样本。
- 4) 每个新样本刚进入 Replay Buffer 时，它的优先级会被设置为无穷大，直至它被输入到网络中进行训练，它的优先级才会得到更新。

2.3 实验

在本节中，将详细报告实验情况，该实验完全基于模拟。在代码实现方面，参考了 Clean Reinforcement Learning Library (CleanRL) 提供的^[40] 编程风格。

2.3.1 变量分布

案例随机生成的概率分布可以描述为：

$$\rho_g \sim U(0, 20), \alpha_g \sim U(-\pi, \pi), \delta \sim U(-\pi, \pi),$$

$$\rho_o \sim U(0, 15), \alpha_o \sim U(-\pi, \pi), r_o \sim U(1.5, 6),$$

$$g = (-\rho_g \sin(\alpha_g), \rho_g \cos(\alpha_g), \delta),$$

$$o = (-\rho_o \sin(\alpha_o), \rho_o \cos(\alpha_o), r_o)$$

程序可以检查障碍物是否与汽车或目标重叠。汽车的宽度为 3 米，长度为 6 米。程序可以确保它们之间至少有 1.5 米的间隙。

2.3.2 结果

测试序号	障碍物存在	是否忽视	精度
1	不存在	否	99% ~ 100%
2	存在	否	99.7% ~ 100%
3	存在	是	76% ~ 80%

表 2-1 Evaluation of the model

在这里使用的性能指标是实现目标的准确性。只有当汽车的位置和方向足够接近时，目标才被判断为达成。没有使用本轮的奖励分数，因为环境是可变的。在评估环境



图 2-3 避障仿真环境和模型运行效果

中记录了一些结果，见表 2-1。需要注意的是，在单个模型中训练了所有情况，因此下面的数字是该模型在不同情况下的表现。

第一个是没有障碍物的版本的准确性，第二个是有障碍物的版本。为了进行一些比较并证明代理在碰撞避免方面学到了东西，进行了另一个评估。让代理在有障碍物的环境中移动，但遮盖了障碍物的观察，这意味着在将观察输入策略之前，将设置 $s_o = (0, 0, 0, 0)$ ，以便代理在移动时可以忽略障碍物。

结果显示了一些有趣的现象。对于测试 1，这是最容易解决的任务，代理可以获得 100% 的准确率，因此在没有障碍物的情况下非常可靠。然而，在测试 3 中，尽管代理拥有测试 1 的可靠策略，但由于缺乏障碍物的观察，其准确率降低到 80%，因为它没有足够的信息来进行判断和决策。当在测试 2 中恢复障碍物的观察时，其准确率再次提升到 99.7%。因此，有充分的理由相信代理已经学会了关于碰撞避免的有用技能。

神经网络是一种近似方法，而控制任务通常需要高精度的动作，99.7% 的准确率是可以接受的。此外，根据综述^[41]，路径规划层对于碰撞避免是不可或缺的，但智能体根本没有任何路径规划模块。还发现另一个有趣的现象是，当代理有两个或更多选项来实现其目标时，它可能会犹豫并长时间停滞不前。原因在于 DDPG 是一种确定性策略，只能处理唯一的选项。如果使用随机策略^[9]，可能会更好，但在稀疏奖励的情况下，最大化熵的效果可能并不明显。

2.3.3 训练曲线

在训练过程中记录了训练中的一些统计数据。在图 2-6b 中，”ddpg-free”是没有障碍物时的准确率，可以看到代理学习得非常快，并在很短的时间内达到了 100%。在图 2-6b 中，是有障碍物时的准确率，增长相对较慢。

图 2-4 显示了单个 Q 网络的损失，而 TD3 有两个 Q 网络。图 2-5 显示了 Q 值，200

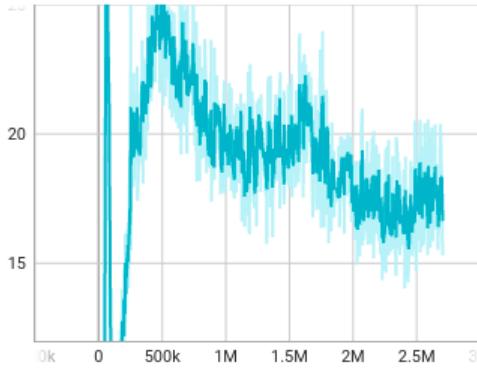


图 2-4 Q 网络损失函数

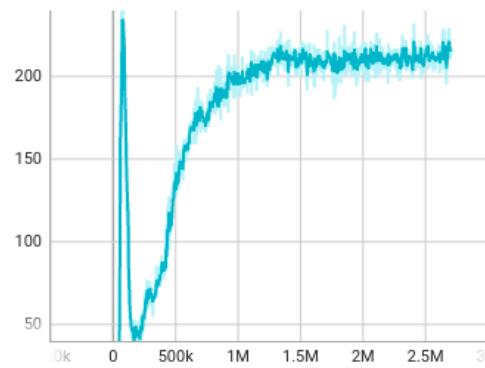


图 2-5 Q 网络输出均值

是一个合理的值，因为 400 的奖励经过大约 100 个时间步折扣后应该接近 200。可以看到，使用 Q 剪切归一化后，损失保持在合理范围内，训练过程平稳。在训练过程中，发现较大的批量大小在强化学习中可能不会导致更好的效果，这确实是一个有趣的现象。一些超参数如下。

random seed = 1	torch deterministic = True	cuda = True
total timesteps = 2.5M	learning rate = 3×10^{-4}	replay buffer size = 10^6
discount rate $\gamma = 0.99$	TD3 update rate $\tau = 0.005$	batch size = 512
learning starts = 25000	TD3 policy update frequency = 2	DDPG exploration noise = 0.1 ,
Q layers = 4	Q hidden = 512	actor layers = 4
actor hidden = 512	activation function = relu	distributed training = False

在应用了前面提到所有的方法之后，就可以训练出效果理想的模型。还进行了一系列消融实验，并把它总结成了一个表格（图 2-6）。实验的结论是，为了让模型收敛，“reward clip”是必要的，如果没有“reward clip”所有方案都无法收敛。而“ $y'_t = \max(y_t, \sum_{i=t}^{\infty} \gamma^{i-t} r_i)$ ”和 PER 的其中至少添加一项即可在该任务上收敛，如果仅仅只有“reward clip”也是无法收敛的。这些实验表明所提出的方案都是切实有效的。图 2-6b 可以看到训练的曲线。

2.4 总结与讨论

利用深度强化学习（DRL）方法解决车辆避障问题，尤其是碰撞避免问题，是无人驾驶技术中的基础和关键任务之一。本章在传统的 HER（Hindsight Experience Replay）方法的基础上，提出了一种新的方法，通过引入障碍物的概念，改进了原有的训练策略。

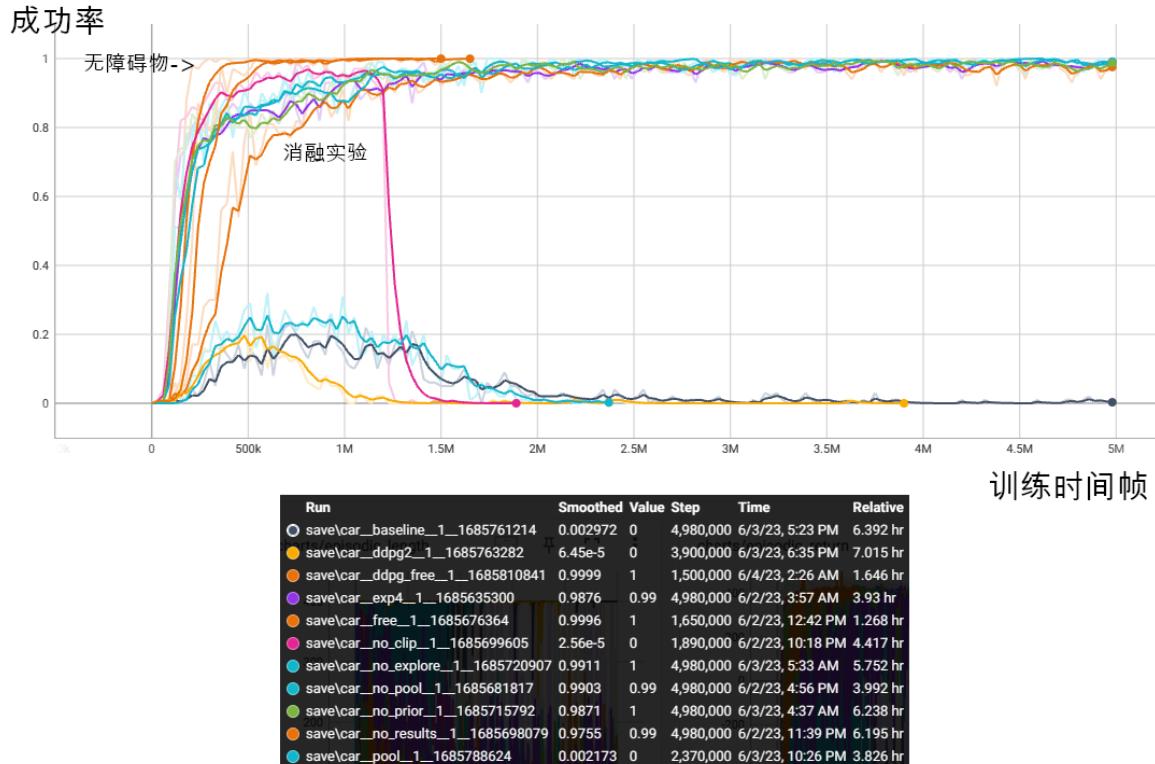
这一创新使得智能体能够在动态环境中不仅关注目标的达成，还能有效规避障碍物，提高了模型在复杂避障任务中的鲁棒性和适应性。

在此基础上，本章还探讨了在可配置环境中训练智能体时采用的一些策略，如设计特定任务的探索策略和应对稀疏奖励的技巧。这些策略的提出表明，针对特定任务定制化的探索方法具有显著优势。同时，本章强调了在强化学习中超越传统马尔可夫决策过程（MDP）的局限性，探索更多与环境互动的方式，以更真实地模拟自然界智能体的学习过程。自然界中的智能体不仅仅是从环境反馈中学习，还可能通过比较过去的经验、尝试其他任务的子技能或技能来增强其适应性，这一点为强化学习中探索和交互的深入研究提供了新的视角。

此外，深度强化学习中的收敛性问题一直是一个挑战，尤其是在复杂的避障任务中。为了确保模型的稳定收敛，本章采用了多种技巧和优化方法，验证了前述的随机离散模型在处理多样化任务和复杂奖励函数时的有效性。该模型表现出较高的鲁棒性，能够在不同的环境条件下稳定运行，为解决复杂的避障任务提供了新的思路和方法。

编号	障碍物	clip	priority	$y'_t = \max(y_t, \sum_{i=t}^{\infty} \gamma^{i-t} r_i)$	探索时固定模型	收敛	准确率
0	×	×	×	×	×	×	0%
1	×	✓	×	×	×	✓	99%
2	✓	✓	×	×	×	×	0%
3	✓	✓	×	×	✓	×	0%
4	✓	✓	✓	✓	✓	✓	99%
5	✓	×	✓	✓	✓	×	0%
6	✓	✓	×	✓	✓	✓	99%
7	✓	✓	✓	×	✓	✓	99%

(a) 消融实验表格



(b) 避障任务训练曲线

图 2-6 局部避障

第3章 面向机器人步态控制与越障的离散化强化学习

3.1 背景

在机器人步态控制和越障任务中，智能体不仅需要完成基本的动作控制，还要应对复杂的地形和障碍物。这些任务通常具有高度的不确定性和动态性，需要智能体在不规则环境中进行高效的路径规划和控制。特别是在越障任务中，机器人往往需要执行一系列连续的、高度精细的动作，这些动作并非简单的线性选择，而是形成多峰分布，即在某些情况下，可能存在多个合适的控制策略或动作组合来达到目标。

多峰分布指的是在特定任务中，不同的动作可能会对应不同的回报或成功路径。例如，在越障任务中，机器人可能通过不同的跳跃方式或通过多次微调步态来克服障碍物，这些方式虽然各自独立，但都能达到成功越障的目的。为避免多个策略选项之间发生干扰，离散化动作空间的强化学习方法成为一种可行的解决方案。通过对动作空间进行离散化，智能体可以更精确地探索和评估每个可能的动作，从而选择最合适的控制策略。而在多峰分布的环境中，离散化的强化学习方法能够更好地适应不同的策略，避免过度集中在单一动作或路径上，从而提高在复杂环境中的适应能力和鲁棒性。这为处理机器人越障和步态控制等任务提供了新的思路和方法。

强化学习（RL）通常对离散任务采用离散动作空间，对连续任务采用连续动作空间。对于离散任务，例如 Atari 游戏，Q 学习及其变体只会详尽评估少量动作。对于连续任务，例如运动控制任务，评估所有可能的动作是不可行的，因此具有离散动作空间的 RL 模型可能会受到维度灾难的影响^[42,43]。为了避免这个问题，针对连续任务的模型直接输出连续值的动作^[9,17,19,43,44]。

然而，在控制系统的背景下，使用离散信号进行控制是广泛采用的。例如，脉宽调制（PWM）用于电动机控制和灯光控制。PWM 用 0 和 1（有时也用 -1）表示信号，属于纯离散信号。然而，通过改变离散值呈现的时间

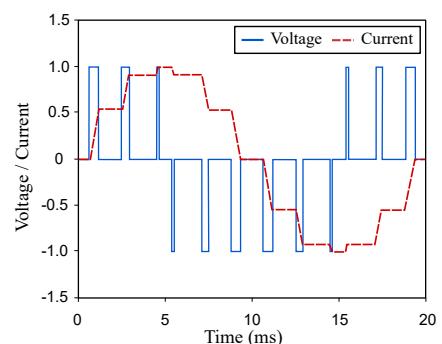


图 3-1 离散电压会产生持续电流

比例，PWM 可以近似连续信号（图 3-1）^[45]，并能够精确控制电动机^[46]。对于发光二极管（LED），由于其光强与输入电压之间的非线性关系，使用连续电压进行控制比使用 PWM 控制的电压更困难^[47]。PWM 通过开关灯光的方式线性化了这种关系，并通过时间比例来控制亮度，从而简化了控制的复杂性。因此，离散信号已被证明对控制任务非常有效，尽管添加离散的约束可能会错过最优的控制律。

基于此，具有离散化动作空间的强化学习模型是否可以在连续任务上表现得与连续模型相似？在对这一想法进行探索后，提出了一种针对连续任务的离散化动作空间模型。

设计了一个离散化 Actor 网络，修改了 Categorical DQN (C51)^[1]，并提出了受到 TD3^[17] 启发的双重离散 Critic 网络，将它们组合在一起。Actor 输出来自均匀离散化动作的动作原子的概率。C51 是值分布强化学习 (Distributional RL) 的一个例子，可以用于建模部分可观测问题的奖励回报分布。它采用离散价值分布，并更新价值类别的概率，而不是拟合曲线，因此能够抵御强烈的奖励影响。

该模型能够解决各种连续任务，其性能接近当前最先进的 (SOTA) 模型。

在 BipedalWalkerHardcore-v3 中，本文采用离散价值分布的模型在离散化动作空间下获得了高于 SOTA 模型的得分。该模型在 10,000 次试验中取得了 324.8 的平均得分，目前（2024 年 5 月）在排行榜上是最高的^[48]。

本节涵盖了与该模型相关的先前工作和基本概念。本文中的数学符号与^[1] 中使用的符号保持一致。为方便起见，表 3-1 提供了符号参考。

符号	描述	典型值
γ	折扣因子。	0.98 或 0.99
V_{MAX}	离散值的上界。	$\frac{1}{1-\gamma}$
V_{MIN}	离散值的下界。	$-\frac{1}{1-\gamma}$
Z	离散值的随机变量。	
z_i	Z 的第 i^{th} 离散值位点。	$[V_{MIN}, V_{MAX}]$
x	当前状态观测的样本。	

a	动作样本向量。	
\hat{a}	动作向量的分布。	
\hat{A}	多维离散化动作空间的动作分布矩阵，其中每行的总和为 1。	
A	动作样本矩阵。 A 中的每一行从 \hat{A} 行中采样。	
\hat{A}'	下一个状态的动作分布。	
r	来自环境的奖励。	≤ 1
x'	下一个状态观测的样本。	
\hat{x}'	下一个状态观测的分布。	
$\hat{\cdot}$	随机变量的分布。	
\cdot'	下一个时间步的变量。	
$\stackrel{D}{=}$	表示定义。	
\mathcal{X}	状态观测的空间。	
\mathcal{A}	动作的空间。	
$\left\lceil \frac{1}{1-\gamma} \right\rceil$	$\frac{1}{1-\gamma}$ 的上限。	
N	Z 的离散位点数量。	51
$\hat{T}Z$	$r + \gamma Z'$ 。	
$\Phi \hat{T}Z$	将 $\hat{T}Z$ 投影回原始离散值位点。	
\tilde{Z}	从孪生 Critic 网络估计的 Z 。	
Θ	离散分布 Critic 网络。	
$(\cdot)_i$	向量的第 i^{th} 元素。	
π	用于动作选择的策略。	
Q	期望标量 Critic 网络。	
ψ_1, ψ_2	第一和第二 Critic 网络的参数。	
ϕ	Actor 网络的参数。	
ψ'_1, ψ'_2, ϕ'	延迟更新的目标网络的参数。	
\leftarrow	表示参数更新。	
$\nabla_{\omega} J$	关于 ω 的 J 的梯度。	

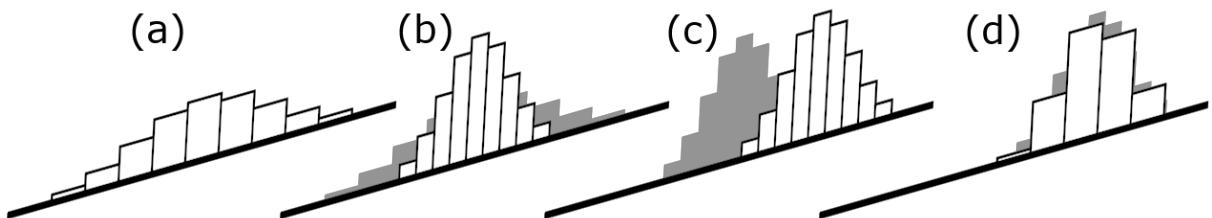
α	学习率。	$\leq 10^{-3}$
B	批量大小。	256 或 512
$t \sim \mathcal{D}$	从重放缓冲区采样。	
n	动作的维度数量。	≤ 20
m	每个动作维度的位点数量。	51
$\mathcal{H}(\hat{A})$	动作 \hat{A} 的熵。	$\leq n \log m$
$\bar{\mathcal{H}}$	动作的最大熵。	$n \log m$
h	动作熵的缩放因子。	0.5
β	探索的系数。	0.5
\sup	表示上界。	

表 3-1 数学符号

3.1.1 离散价值分布

在环境中的随机转移过程 $(\mathbf{x}, \mathbf{a}) \rightarrow (\hat{\mathbf{x}}', \hat{\mathbf{a}}')$ 中， \mathbf{x} 代表环境的当前观察状态， \mathbf{a} 指定了针对 \mathbf{x} 采取的动作。由此产生的状态分布用 $\hat{\mathbf{x}}'$ 表示。随机策略输出一个动作分布 $\hat{\mathbf{a}}'$ ，在任务中实际采取的动作 \mathbf{a}' 将从 $\hat{\mathbf{a}}'$ 中进行采样。

与过程 $(\mathbf{x}, \mathbf{a}) \rightarrow (\hat{\mathbf{x}}', \hat{\mathbf{a}}')$ 相关的价值 Z 使用递归方程进行表示： $Z(\mathbf{x}, \mathbf{a}) \stackrel{D}{=} R(\mathbf{x}, \mathbf{a}) + \gamma Z(\hat{\mathbf{x}}', \hat{\mathbf{a}}')$ ，其中 $R(\mathbf{x}, \mathbf{a})$ 表示环境的随机奖励函数， γ 表示折扣率。

图 3-2 C51 更新 Z 的操作。改编自^[1]。

在 Categorical DQN^[1] 中，值 Z 被概念化为具有离散值分布的随机变量。为了更新分布，Bellemare 等人^[1] 提出了一个更新规则（图 3-2），分为四个步骤：(a) Z 的当前价值分布。(b) 折扣因子 γ 收缩分布位点的间隔。(c) 当前奖励 R 使得分布位点发生移动。

(d) 结果分布 $R + \gamma Z$ 通过 Φ 映射回分布位点。离散分布位点的数量 $N \in \mathbb{N}$ (整数集) 表示值域所需的离散化粒度, 边界 $V_{MIN}, V_{MAX} \in \mathbb{R}$ (实数集) 分别指定值的下限和上限。离散分布位点的集合构建为 $\{z_i = V_{MIN} + (i - 1) \Delta z | i = 1, 2, \dots, N\}$, 其中间隔 Δz 通过 $\frac{V_{MAX} - V_{MIN}}{N-1}$ 计算。每个离散分布位点出现的概率是通过神经网络 $\Theta : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}^N$ 确定的 (\mathcal{X} 表示状态定义域, \mathcal{A} 表示动作的定义域), 即

$$Z(\mathbf{x}, \mathbf{a} | \Theta) = z_i \quad w.p. \quad p_i(\mathbf{x}, \mathbf{a}) = \frac{e^{(\Theta(\mathbf{x}, \mathbf{a}))_i}}{\sum_j^N e^{(\Theta(\mathbf{x}, \mathbf{a}))_j}}$$

对于随机转移的元组 $\mathbf{t} = (\mathbf{x}, \mathbf{a}, r, \mathbf{x}')$, 对每个离散分布位点 z_j 应用 Bellman 更新, 表示为 $\hat{T}z_j := r + \gamma z_j$ (r 表示奖励)。与 $\hat{T}z_j$ 相关的概率 $p_j(\mathbf{x}', \pi(\mathbf{x}'))$ 随后在相邻的离散分布位点之间重新分配。投影后的离散概率分布记为 $\Phi\hat{T}Z(\mathbf{x}, \mathbf{a} | \Theta)$, 它的第 i 个元素为:

$$P(\Phi\hat{T}Z(\mathbf{x}, \mathbf{a} | \Theta) = z_i) = \sum_{j=1}^N \left[1 - \frac{|[\hat{T}z_j]_{V_{MIN}}^{V_{MAX}} - z_i|}{\Delta z} \right]_0^1 p_j(\mathbf{x}', \pi(\mathbf{x}')) \quad (3-1)$$

符号 $[\cdot]_a^b$ 表示该值被限制在区间 $[a, b]$ 内。有关这些概念的详细阐述和验证, 请参阅原始论文^[1]。

3.1.2 孪生 Critic 网络

TD3^[17] 采用孪生 Critic 网络学习来减少 Q 值的高估偏差。TD3 由三个网络组成: $Q_{\psi_1}(\mathbf{x}, \mathbf{a})$ 、 $Q_{\psi_2}(\mathbf{x}, \mathbf{a})$ 和 $\pi_\phi(\mathbf{x})$ 。这两个 Q 网络 $Q_{\psi_1}(\mathbf{x}, \mathbf{a})$ 和 $Q_{\psi_2}(\mathbf{x}, \mathbf{a})$ 虽然最初具有不同的参数, 但会同时使用相同的学习信号进行训练。 $\pi_\phi(\mathbf{x})$ 则作为 Actor 网络。TD3 的数据批次更新过程包括三个步骤:

$$y \leftarrow r + \gamma \min_{i=1,2} Q_{\psi'_i}(\mathbf{x}', \pi_{\phi'}(\mathbf{x}') + \epsilon)$$

首先, 利用孪生网络对奖励回报进行更精确的估计, 记作 y 。其中 ψ'_i 是第 i 个目标 Critic 网络的延迟更新参数, ϕ' 是目标 Actor 网络的延迟更新参数, ϵ 是微小噪声。

$$\psi_i \leftarrow \psi_i - \frac{\alpha}{B} \sum_{t \sim \mathcal{D}} \nabla_{\psi_i} (y - Q_{\psi_i}(\mathbf{x}, \mathbf{a}))^2$$

然后应用均方误差损失函数，使双重 Critic 网络与修正后的 Q 值 y 对齐，从而第 i 个 Critic 网络的参数 ψ_i 。其中 α 是学习率， B 是样本批次大小， $t \sim \mathcal{D}$ 表示来自重放缓冲区的一批转移样本， $t = (\mathbf{x}, \mathbf{a}, r, \mathbf{x}')$ 。

$$\phi \leftarrow \phi + \frac{\alpha}{B} \sum_{t \sim \mathcal{D}} \nabla_\phi \pi_\phi(\mathbf{x}) \nabla_\mathbf{a} Q_{\psi_1}(\mathbf{x}, \pi_\phi(\mathbf{x}))|_{\mathbf{a}=\pi_\phi(\mathbf{x})}$$

最后，Actor 网络 $\pi_\phi(\mathbf{x})$ 根据 $Q_{\psi_1}(\mathbf{x}, \mathbf{a})$ 独立于 $Q_{\psi_2}(\mathbf{x}, \mathbf{a})$ 进行学习，其中 ϕ 表示 Actor 网络的参数。

3.2 模型与方法

模型（图 3-3）通过解决之前工作的局限性，扩展了算法的性能和鲁棒性，采用了多维离散化动作空间、剪切双重 Q 学习以处理离散值分布、对应的 Critic 和 Actor 学习规则，以及平衡探索与利用。

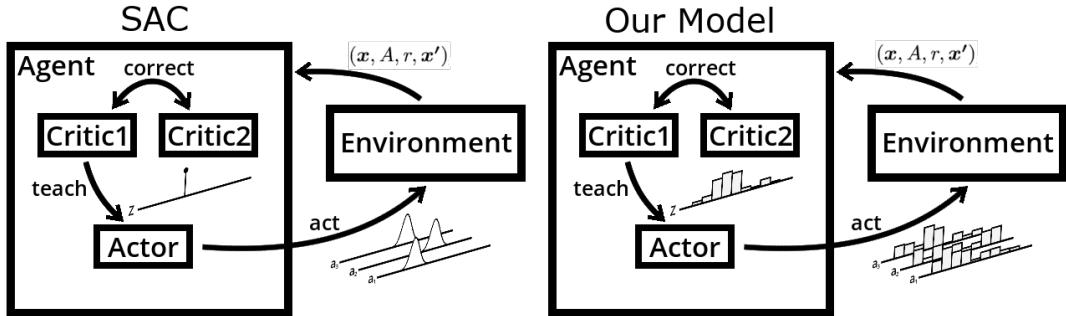


图 3-3 算法框架总览

3.2.1 多维离散化 Actor

模型基于多维离散化动作空间。一个一维连续动作空间被离散化为 m 个离散化动作位点 $\{a_1, a_2, \dots, a_m\}$, $m \in \mathbb{N}$ ，其中 \mathbb{N} 表示自然数集合。然后，将离散化应用于 n 维连续动作空间的每个维度，因此在这个新的离散空间 \mathcal{A} 中的一个动作可以表示为一个矩阵。

$A \stackrel{D}{=} [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]^\top$ ，其中每一行是对应动作维度的独热（one-hot）编码。这种形状方便与动作概率分布 \hat{A} 匹配， \hat{A} 将作为策略网络的输出，并且 \hat{A} 的每一行的和为 1。

当从 \hat{A} 中采样 A 时, A 中的每一行是从 \hat{A} 中对应行的概率中采样得来的。

在这个动作空间中, 存在 m^n 个离散潜在动作。考虑到如此巨大的搜索空间, 采用传统 DQN 算法中使用的穷举搜索方法来寻找最大 Q 值是不可行的。在本研究中, 提出通过利用动作概率矩阵来对智能体在动作空间 \mathcal{A} 中的随机行为建模, 因此将 Actor 设定为 $\pi : \mathcal{X} \rightarrow \mathbb{R}^{n \times m}$, 即:

$$\pi(\mathbf{x}) \stackrel{D}{=} \begin{bmatrix} p_{11}(\mathbf{x}) & p_{12}(\mathbf{x}) & \cdots & p_{1m}(\mathbf{x}) \\ p_{21}(\mathbf{x}) & p_{22}(\mathbf{x}) & \cdots & p_{2m}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1}(\mathbf{x}) & p_{n2}(\mathbf{x}) & \cdots & p_{nm}(\mathbf{x}) \end{bmatrix}, \quad (3-2)$$

其中, \mathbf{x} 是观察到的状态, $\sum_{j=1}^m p_{ij}(\mathbf{x}) = 1$ for $i = 1, 2, \dots, n$, $p_{ij}(\mathbf{x}) \geq 0$ 。

这个 π 描述了一个随机的多维离散化 Actor。后面的部分将详细说明如何使用神经网络来近似 π 。请注意, 在原始的连续动作空间中, 动作维度是独立的, 因此在 A 中, 行与行之间的元素也是独立的。

3.2.2 孪生离散价值分布对齐裁剪

之前在 3.1.2 节中讨论了 TD3 通过孪生 Q 学习来减轻高估的方法。在这里, 对其进行了修改, 以适应离散值分布。

孪生 Q 学习使用两个 Critic 网络 $\Theta_{\psi_1}(\mathbf{x}, \hat{A})$ 和 $\Theta_{\psi_2}(\mathbf{x}, \hat{A})$, 以及一个 Actor 网络 $\pi_\phi(\mathbf{x})$ 。它还具有目标网络 $\Theta_{\psi'_1}(\mathbf{x}, \hat{A})$ 、 $\Theta_{\psi'_2}(\mathbf{x}, \hat{A})$ 和 $\pi_{\phi'}(\mathbf{x})$, 以确保训练的稳定性。上面的下标 $\psi_1, \psi_2, \phi, \psi'_1, \psi'_2$ 和 ϕ' 表示相应网络的参数。给定一个转移元组 $t = (\mathbf{x}, A, r, \mathbf{x}')$, 考虑如何有效利用这些目标网络来生成价值分布新的估计 $\Phi \hat{T} \tilde{Z}(\mathbf{x}, \hat{A} | \Theta_{\psi'_1}, \Theta_{\psi'_2})$ 。使用 $\Theta_{\psi'_1}$ 和 $\Theta_{\psi'_2}$, 对于 $\hat{A}' = \pi_{\phi'}(\mathbf{x}')$, 得到 $\Phi Z(\mathbf{x}', \hat{A}' | \Theta_{\psi'_i})$:

$$P(Z(\mathbf{x}', \hat{A}' | \Theta_{\psi'_i}) = z_k) \stackrel{D}{=} \frac{e^{(\Theta_{\psi'_i}(\mathbf{x}, \hat{A}'))_k}}{\sum_j^N e^{(\Theta_{\psi'_i}(\mathbf{x}, \hat{A}'))_j}} \quad (3-3)$$

孪生 Critic 网络和离散值分布的计算过程 (如图 3-4 所示)。(a) 首先, 两个 Critic 网络分别根据 \mathbf{x}' 估计离散值分布。(b) 其次, 分别累加这些分布。(c) 然后, 对于累积分布中的

每个类别，选择概率更高的类别形成新的累积分布，从而使得值不易受到高估的影响。

(d) 最后，新的累积分布中每个类别（除了第一个）减去前一个类别，将其映射回离散值分布。

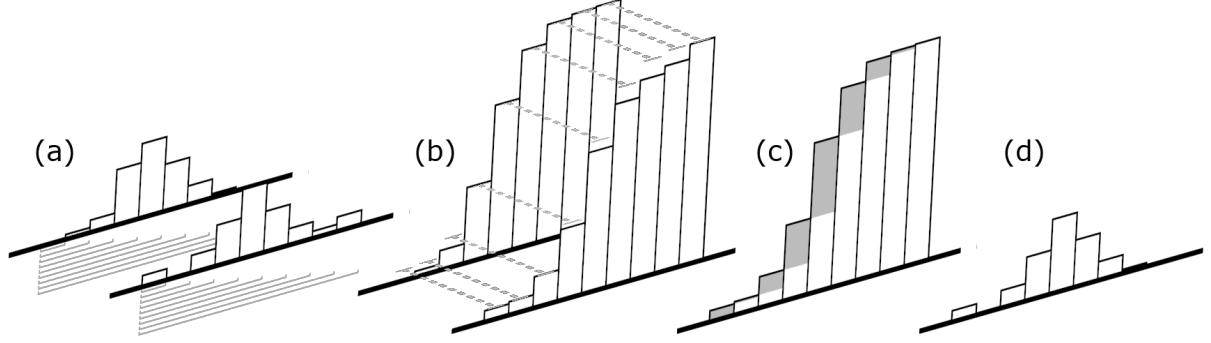


图 3-4 孪生 Critic 网络和离散值分布及其更新方法示意

在过程 (b) 中，累加了 (a) 中的分布。对于离散值分布 Z ，

$$P(Z \in \{z_1, z_2, \dots, z_k\}) = \sum_{j=1}^k P(Z = z_j) \quad (3-4)$$

对于第 k 个值位点，第三和第四个步骤可以表示为：

$$P(\tilde{Z}(x', \hat{A}' | \Theta_{\psi'_1}, \Theta_{\psi'_2}) = z_k) = \begin{cases} c_k & \text{if } k = 1 \\ c_k - c_{k-1} & \text{if } k > 1 \end{cases} \quad (3-5)$$

其中 $c_k = \max_{i=1,2} P(Z(x', \hat{A}' | \Theta_{\psi'_i}) \in \{z_1, z_2, \dots, z_k\})$ 。对于转移样本 $\mathbf{t} = (\mathbf{x}, A, r, \mathbf{x}')$ 和第 i 个值位点，Bellman 操作如下：

$$P(\Phi \hat{\mathcal{T}} \tilde{Z}(\mathbf{x}, A | \Theta_{\psi'_1}, \Theta_{\psi'_2}) = z_i) = \sum_{j=1}^N \left[1 - \frac{|[\hat{\mathcal{T}} z_j]_{V_{MIN}}^{V_{MAX}} - z_i|}{\Delta z} \right]_0^1 P(\tilde{Z}(\mathbf{x}', \hat{A}' | \Theta_{\psi'_1}, \Theta_{\psi'_2}) = z_j) \quad (3-6)$$

其中 A 表示实际采取的行动，而不是概率分布， $\Phi \hat{\mathcal{T}} \tilde{Z}(\mathbf{x}, A | \Theta_{\psi'_1}, \Theta_{\psi'_2})$ 是用于训练 Θ_{ψ_1} 和 Θ_{ψ_2} 的修正离散值分布。

第 3.2.1 节介绍了多维离散化 Actor。Actor 通过概率矩阵输出动作分布。这个矩阵

作为 Critic 网络动作组件的直接输入，使其能够建模服从多峰分布的随机动作的影响。结合以上讨论，Critic 网络对大小为 B 的数据批次的更新规则定义为：

$$\begin{aligned} Z_1 &\stackrel{D}{=} \Phi \hat{\mathcal{T}} \tilde{Z}(\mathbf{x}, A | \Theta_{\psi'_1}, \Theta_{\psi'_2}), \quad Z_2 \stackrel{D}{=} Z(\mathbf{x}, \hat{A} | \Theta_{\psi_i}) \\ \boldsymbol{\psi}_i &\leftarrow \boldsymbol{\psi}_i - \frac{\alpha}{B} \sum_{t \sim D} \nabla_{\boldsymbol{\psi}_i} D_{KL}(Z_1 || Z_2) \end{aligned} \quad (3-7)$$

其中 D_{KL} 表示 KL 散度, α 表示学习率，进一步地，

$$\nabla_{\boldsymbol{\psi}_i} D_{KL}(Z_1 || Z_2) = - \sum_{j=1}^N P(Z_1 = z_j) \nabla_{\boldsymbol{\psi}_i} \log P(Z_2 = z_j)$$

在这个方程中，消除了与 $\boldsymbol{\psi}_i$ 无关的项，从而获得了与交叉熵损失一致的形式。 Z_1 表示从孪生 Critic 网络获得的新值分布估计， Z_2 是 Critic 网络的输出结果。通过这种方式，每个 Critic 的输出被调整为与修正后的值 Z_1 对齐，从而减少了高估偏差。

训练 Critic 网络时，使用交叉熵损失拟合分类分布，比直接用均方误差损失拟合曲线更稳定。这种现象可能是因为分类分布中的概率限制在 0 和 1 之间，而奖励本身的大可以相差一百倍。在 Bipedalwalker 任务中，摔倒会导致巨大的惩罚 (-100)，这会影响在行走期间 Critic 的学习（通常在 -0.3 到 0.3 之间）。在拟合这种多峰分布时，每个峰值对其他峰值的学习产生类似的影响，就像离群点一样。

3.2.3 策略学习

假设任务中的动作分布不是单峰分布，特别是在像 BipedalWalkerHardcore-v3 这样具有变化地形的任务中。因此，按照第 3.2.1 节中的描述对动作空间进行离散化。为了稳健地训练 Actor，使用与第 3.2.2 节中介绍的训练 Critic 网络相似的损失函数来训练 Actor，从而有助于模型的整体稳定性。

与其他具有 Actor-Critic 架构的强化学习模型类似，Actor 的更新旨在最大化 Critic 网络预测的 Q 值。不同的是，在本模型中，Critic 网络的输出是概率，因此可以将累积

分布作为目标。更具体地，对于第 k 个值位点，

$$\begin{aligned} P(Z(\mathbf{x}, \pi_\phi(\mathbf{x}) | \Theta_{\psi_1}) \in \{z_1, z_2, \dots, z_k\}) &\rightarrow 0 \\ P(Z(\mathbf{x}, \pi_\phi(\mathbf{x}) | \Theta_{\psi_1}) \in \{z_{k+1}, z_{k+2}, \dots, z_N\}) &\rightarrow 1. \end{aligned} \quad (3-8)$$

这里的符号“ \rightarrow ”表示向某个值的趋向。目标是使得策略在低值原子上最小化 Z 出现的概率，同时在高值原子上最大化该概率。通过应用二元交叉熵损失，策略学习规则如下建立：

$$\phi \leftarrow \phi + \frac{\alpha}{B} \sum_{t \sim \mathcal{D}} \sum_{j=1}^N \nabla_\phi [0 \log \rho_j + 1 \log(1 - \rho_j)] = \phi + \frac{\alpha}{B} \sum_{t \sim \mathcal{D}} \sum_{j=1}^N \nabla_\phi \log(1 - \rho_j) \quad (3-9)$$

其中， α 表示学习律，且

$$\rho_j \stackrel{D}{=} P(Z(\mathbf{x}, \pi_\phi(\mathbf{x}) | \Theta_{\psi_1}) \in \{z_1, z_2, \dots, z_j\}) \quad (3-10)$$

通过该规则，可以把策略调整到最大化高价值回报位点发生概率的选项。

3.2.4 探索与利用

策略改进是 Actor 学习的基本组成部分；同样，利用也发挥着重要作用。本研究引入了一种受失败驱动的启发式探索方法，利用伤害感知信息来增强策略探索策略。

定义 3.1 给定一个动作分布 $\hat{A} = \pi(\mathbf{x})$ ，动作熵定义为：

$$\mathcal{H}(\hat{A}) \stackrel{D}{=} - \sum_{i=1}^n \sum_{j=1}^m p_{ij}(\mathbf{x}) \log p_{ij}(\mathbf{x}) \quad (3-11)$$

此外， $\mathcal{H}(\hat{A})$ 有一个可计算的上界：

$$\overline{\mathcal{H}} \stackrel{D}{=} n \log m \geq \mathcal{H}(\hat{A}) \quad \forall \pi : \mathcal{X} \rightarrow \mathbb{R}^{n \times m} \quad (3-12)$$

模型的目标是将动作熵与置信水平相关联。具体来说，在离散值分布的低离散位点

处发生的概率较高时，增加动作熵 $\mathcal{H}(\hat{A})$ 。为此，引入了一个熵探索项。提议的 Actor 更新规则如下：

$$\begin{aligned}\phi &\leftarrow \phi + \frac{\alpha\beta}{B} \sum_{t \sim \mathcal{D}} s \nabla_\phi \frac{\mathcal{H}(\pi_\phi(\mathbf{x}))}{\bar{\mathcal{H}}} \\ s &= \begin{cases} 1 & \text{if } \max_{1 \leq j \leq N} \frac{N-j}{N-1} h \rho_j \geq \frac{\mathcal{H}(\pi_\phi(\mathbf{x}))}{\bar{\mathcal{H}}} \\ 0 & \text{otherwise} \end{cases} \quad (3-13)\end{aligned}$$

其中， ρ_j 与公式 (3-10) 中相同， $\beta > 0$ 是熵项的系数， $0 < h \leq 1$ 调节动作熵的规模。每个离散位点都有一个动作熵阈值 $\frac{N-j}{N-1} h \rho_j$ ，只有当动作熵 $\mathcal{H}(\pi_\phi(\mathbf{x}))$ 降低到该阈值以下时，熵探索项才会激活。随着 j 的增加，这个阈值降低，意味着更高值的原子具有更低的阈值。

使用累积分布 ρ_j 来表示代理在当前状态 \mathbf{x} 下的置信水平。需要注意的是，对于高置信度代理的第 j 个值位点， ρ_j 应该是一个较小的标量，因为它表示第 1 个位点和第 j 个位点之间的概率，这代表了较低的值范围。使用 ρ_j 来修正动作熵 $\mathcal{H}(\pi_\phi(\mathbf{x}))$ ，因此低置信度代理会增加动作熵，以寻求针对状态 \mathbf{x} 的多种解决方案，而高置信度代理则不会。

将此与前一节的内容结合起来，Actor 的综合更新规则为：

$$\phi \leftarrow \phi + \frac{\alpha}{B} \sum_{t \sim \mathcal{D}} \sum_{j=1}^N \nabla_\phi \log(1 - \rho_j) + \frac{\alpha\beta}{B} \sum_{t \sim \mathcal{D}} s \nabla_\phi \frac{\mathcal{H}(\pi_\phi(\mathbf{x}))}{\bar{\mathcal{H}}} \quad (3-14)$$

在训练阶段的数据收集过程中，模型还会记录每个动作位点的最近执行时间。在一半的 episode 中，在执行动作之前，对于每个动作维度，以一个给定的小概率，模型将动作位点替换为未执行时间最长的位点。选择的典型概率约为 $0.05/n$ ，其中 n 是动作维度的数量。

3.3 实验

在多个任务的连续控制上测试了本模型，包括 BipedalWalkerHardcore-v3 和 MuJoCo 任务，并评估了该离散模型的性能。选择了 SAC、TD3 和 TQC 作为基准，这些是针对连续任务的主流 off-policy 算法，并将它们应用于相同的任务进行比较。实验在一台配

备 Intel® Core™ i9-12900 处理器、64GB RAM 和 NVIDIA® GeForce RTX™ 4090 的桌面工作站上进行。在代码实现方面，参考了 Clean Reinforcement Learning Library (CleanRL) 提供的^[40] 编程风格。

3.3.1 奶酪陷阱问题

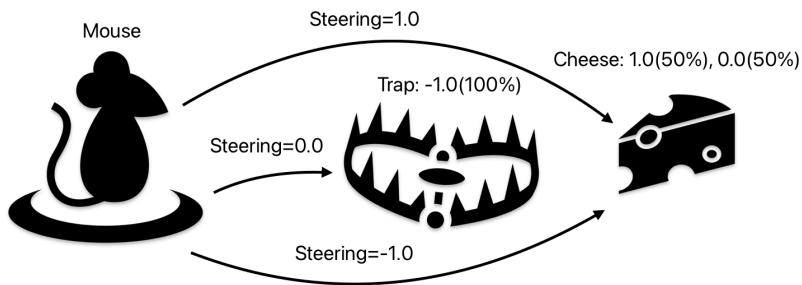


图 3-5 奶酪陷阱问题

设计了一个叫做“陷阱或奶酪”的玩具任务，旨在说明连续模型平均好的动作可能导致一个坏的动作，而本模型没有这个问题。如图 3-5 所示，老鼠前面有一个陷阱，陷阱后面是一块奶酪。当老鼠选择向前移动时，它会掉进陷阱并死亡，得到的奖励是 -1.0。当老鼠选择向左或向右转时，它可以绕过陷阱，抵达奶酪。然而，奶酪有 50% 的概率已经腐坏，无法食用，得到的奖励为 0.0。如果奶酪没有过期，奖励为 1.0。显然，正常的老鼠不会选择走进陷阱。

在这个任务中，结果显示，SAC 往往毫不犹豫地选择中间路线，走进陷阱，平均得分保持在 -1.0。相比之下，本文的离散模型能够学习正确的策略，平均得分保持在 0.5。这个简单的任务对 SAC 来说是困难的，因为尽管它的 Critic 网络可以学习到向前移动是一个非常糟糕的选择，但由于向前移动可以被视为向左和向右移动的平均，SAC 仍然选择向前移动。这个问题可能在倾向于平均最佳动作的连续强化学习模型中普遍存在。在 BipedalWalkerHardcore 任务中，跨越障碍时也具有类似的特性。例如，在面对前方大箱子障碍物时，保守地站在原地和积极地往前跨越都可以避免摔倒，但介于这两个选择之间的动作摔跤的风险就会很大。因此，有理由怀疑这就是为什么连续模型无法像本模型那样解决这个任务的原因。

把奶酪陷阱问题的 Q 函数描述为

$$Q(x_0, a) = \begin{cases} 0.5 & \text{id } a \in [-1 - \delta, -1 + \delta] \cup [1 - \delta, 1 + \delta] \\ -1 & \text{otherwise} \end{cases} \quad (3-15)$$

其中 δ 用来表示可以获取到高奖励的值域的宽度, $0 < \delta < 1$ 。为了方便, 将该区域用符号 $\mathcal{C}(\delta)$ 来表示。现关心的是正态分布 $a \sim N(\mu, \sigma^2)$ 在 $\mathcal{C}(\delta)$ 上的最大似然估计, 在概率密度的对数上进行积分。

$$\begin{aligned} \log L &= \int_{\mathcal{C}(\delta)} \log\left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(a-\mu)^2}{2\sigma^2}}\right) da \\ &= -4\delta \log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2} \int_{\mathcal{C}(\delta)} (a - \mu)^2 da \\ &= -4\delta \log(\sqrt{2\pi}\sigma) - \frac{1}{6\sigma^2} [(1 + \delta - \mu)^3 - (1 - \delta - \mu)^3 + (-1 + \delta - \mu)^3 - (-1 - \delta - \mu)^3] \\ &= -4\delta \log(\sqrt{2\pi}\sigma) - \frac{1}{6\sigma^2} [6(1 - \mu)^2\delta + 2\delta^3 + 6(1 + \mu)^2\delta + 2\delta^3] \\ &= -4\delta \log(\sqrt{2\pi}\sigma) - \frac{2\delta}{3\sigma^2} [3(1 + \mu^2) + \delta^2] \end{aligned} \quad (3-16)$$

令 $\frac{\partial \log L}{\partial \mu} = 0$, $\frac{\partial \log L}{\partial \sigma} = 0$, 可以得到 μ 和 σ 的最大似然估计。

$$\begin{aligned} \frac{\partial \log L}{\partial \mu} &= -\frac{4\delta\mu}{\sigma^2}, \quad \tilde{\mu} = 0 \\ \frac{\partial \log L}{\partial \sigma} &= -\frac{4\delta}{\sigma} + \frac{4\delta}{3\sigma^3} [3(1 + \mu^2) + \delta^2], \quad \tilde{\sigma}^2 = 1 + \frac{\delta^2}{3} \end{aligned} \quad (3-17)$$

验算一下 $\tilde{\mu} = 0, \tilde{\sigma}^2 = 1 + \frac{\delta^2}{3}$ 是否是唯一的极大值点, 首先计算它的 Hessian 矩阵。

$$\begin{bmatrix} \frac{\partial^2 \log L}{\partial \mu^2} & \frac{\partial^2 \log L}{\partial \mu \partial \sigma} \\ \frac{\partial^2 \log L}{\partial \mu \partial \sigma} & \frac{\partial^2 \log L}{\partial \sigma^2} \end{bmatrix} = \begin{bmatrix} -\frac{4\delta}{\sigma^2} & \frac{8\delta\mu}{\sigma^3} \\ \frac{8\delta\mu}{\sigma^3} & \frac{4\delta}{\sigma^4} (\sigma^2 - \delta^2 - 3) \end{bmatrix} \quad (3-18)$$

将 $\tilde{\mu} = 0, \tilde{\sigma}^2 = 1 + \frac{\delta^2}{3}$ 代入, 可以得到

$$\begin{bmatrix} \frac{\partial^2 \log L}{\partial \mu^2} & \frac{\partial^2 \log L}{\partial \mu \partial \sigma} \\ \frac{\partial^2 \log L}{\partial \mu \partial \sigma} & \frac{\partial^2 \log L}{\partial \sigma^2} \end{bmatrix}_{\tilde{\mu}, \tilde{\sigma}} = \begin{bmatrix} -\frac{4\delta}{1 + \frac{\delta^2}{3}} & 0 \\ 0 & -\frac{8\delta}{1 + \frac{\delta^2}{3}} \end{bmatrix} \preceq 0 \quad (3-19)$$

因此 $\tilde{\mu} = 0, \tilde{\sigma}^2 = 1 + \frac{\delta^2}{3}$, 是定义域上唯一的极大值点。尽管 $N(\tilde{\mu}, \tilde{\sigma}^2)$ 是在正态分布假设下的对于集合 $\mathcal{C}(\delta)$ 的最大似然估计, 然而它的最大概率密度点 $\tilde{\mu}$ 在 Q 函数上的取值并不令人满意, 显然 $Q(x_0, \tilde{\mu}) = -1$ 。现在离散分布上再求一次最大似然估计。

$$P(a = a_i) = p_i, \quad i = 1, 2, \dots, m, \quad \sum_i^m p_i = 1.0, \quad p_i \geq 0 \quad (3-20)$$

在离散分布下, 动作 a 的取值范围为 $\mathcal{A} = \{a_1, a_2, \dots, a_m\}$, 并且 $\mathcal{A} \cap \mathcal{C}(\delta) \neq \emptyset$ 。

$$L = \prod_{\mathcal{A} \cap \mathcal{C}(\delta)} p_i \quad (3-21)$$

那么根据均值不等式, 有

$$\sqrt{\prod_{\mathcal{A} \cap \mathcal{C}(\delta)} p_i} \leq \frac{\sum_{\mathcal{A} \cap \mathcal{C}} p_i}{\|\mathcal{A} \cap \mathcal{C}(\delta)\|} \leq \frac{1}{\|\mathcal{A} \cap \mathcal{C}(\delta)\|} \quad (3-22)$$

上面的不等式的两个等号是可以取到的, 因此在离散分布下的最大似然估计为

$$\tilde{p}_i = \begin{cases} \frac{1}{\|\mathcal{A} \cap \mathcal{C}(\delta)\|} & if \quad a_i \in \mathcal{C}(\delta) \\ 0 & otherwise \end{cases} \quad (3-23)$$

在离散分布的最大似然估计下, 取概率最大的一点 a_k , 显然它有, $Q(x_0, a_k) = 0.5$ 。根据上面的讨论, 可以知道, 在应对复杂障碍物时, 离散分布可能会相对于正态分布更有优势, 至少在最大似然估计的尺度下是如此。

3.3.2 BipedalWalkerHardcore-v3

BipedalWalkerHardcore-v3^[21] 任务是 OpenAI Gym 中的基准测试之一。与 OpenAI Gym 中其他运动控制任务相比, BipedalWalkerHardcore-v3 的挑战来自于地形的变化、部分可观察性和摔倒时的高惩罚。该任务的目标是控制一个平面双足机器人在复杂地形中行走, 这些地形包含随机生成的障碍物, 如梯子、树桩和陷阱。智能体必须找到并学习应对每种障碍物的不同动作。机器人的环境观测是通过激光雷达进行的, 返回的是立即地形的 10 个激光测距仪测量值, 因此智能体对环境的观察是部分可观察的。

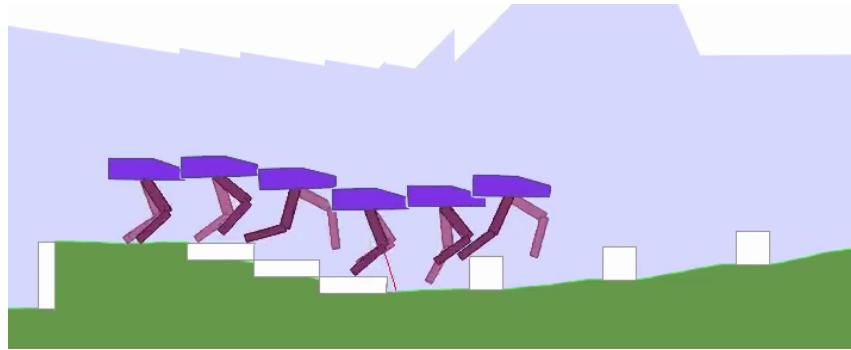


图 3-6 The BipedalWalkerHardcore-v3 任务

该任务还对机器人摔倒给予了较大的惩罚，模拟“伤害刺激”，这干扰了 Critic 网络的收敛，并可能阻碍技能的获取。根据^[49]的说法，将 TD3^[17]适配到 BipedalWalkerHardcore-v3 任务需要减少这些惩罚。

在 BipedalWalkerHardcore-v3 任务上训练了本模型和基准模型，训练了 2000 万个时间步。图 3-7 左侧显示了训练期间的奖励回报。本模型优于基准模型。在评估配置中，本模型在 10,000 次试验中达到了平均得分 324.8。在图 3-7 中展示了本模型和基准模型的训练曲线，并在表 3-2 中总结了它们在 10,000 次试验中的最终评估结果。

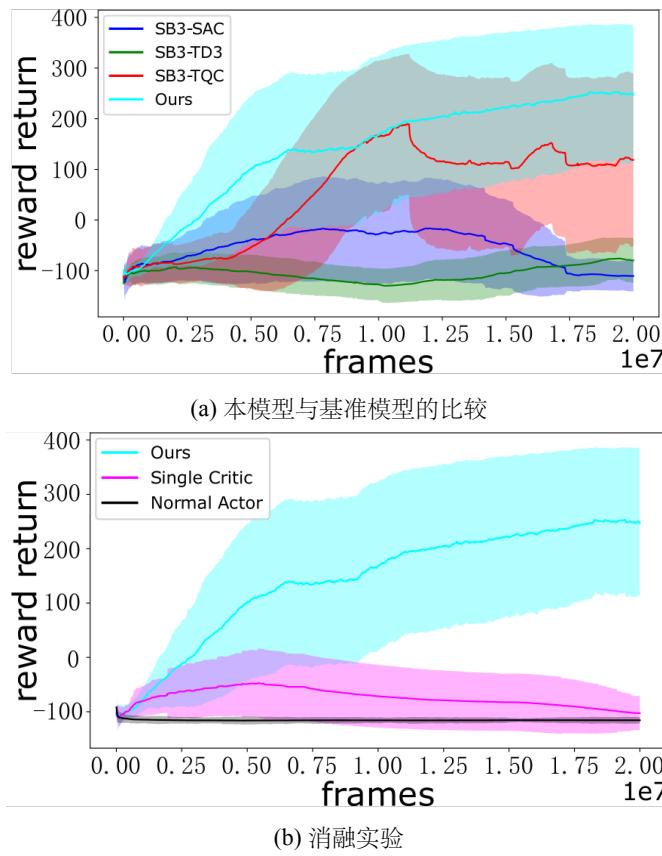


图 3-7 BipedalWalkerHardcore-v3 任务中的训练奖励

表 3-2 BipedalWalkerHardcore-v3 性能评估

Task	Ours	SB3-SAC	SB3-TD3	SB3-TQC
BipedalWalkerHardcore-v3	324.8 ± 31.6	5.1 ± 97.3	-20.1 ± 22.4	217.9 ± 120.3

为了理解每个模块的必要性，在 BipedalWalkerHardcore-v3 任务上进行了消融研究（图 3-7 右侧）。禁用了孪生 Critic 网络，结果通过标记为“Single Critic”的曲线表示。按照 D3PG^[25] 的方法，用基于孪生 Critic 网络的传统连续动作 Actor 替换了离散化 Actor。结果通过标记为“正常 Actor”的曲线表示。这些结果表明，本模型中提出的不同模块对于模型的性能是必要的。

3.3.3 MuJoCo

尽管在构建本模型时，主要在需要冒险行为的强奖励异常值和变化环境的任务上进行了测试，但也需要在更为一致的典型连续控制任务中对其进行评估。实验在 MuJoCo^[21] 中进行，涵盖了一系列任务，包括 Ant、HalfCheetah、Hopper、Humanoid 和 Walker2D。这些任务旨在控制相应类型的机器人进行向前运动。由于环境是平坦的地形，机器人周围环境的观察并不是必需的。因此，任务的状态可以通过任务提供的本体感觉器、里程计和加速度计完全获取，并且这些任务表现出明显的马尔可夫特性。

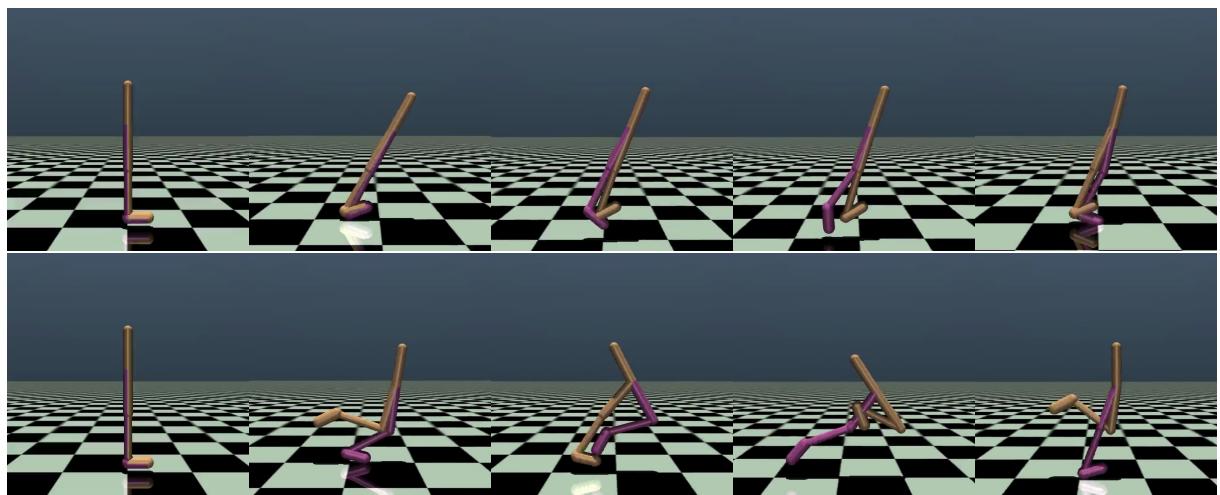


图 3-8 Walker2d 任务对比：上为 TD3 策略，下为本模型策略

表 3-3 MuJoCo 评估。SAC、TD3、TQC 的数据来自^[19]。

Task	Ours	SAC	TD3	TQC
Ant	6988	6160	5680	8010
HalfCheetah	13800	12410	15120	18090
Hopper	4118	2860	3310	3710
Humanoid	9992	7760	5400	9540
Walker2D	5775	5760	5110	7030

虽然将连续动作空间离散化可能会降低控制精度并增加问题解决的难度，但并没有观察到本文提出的模型在 MuJoCo 任务上的性能出现明显的骤降。然而，它的学习效率可能会下降，导致需要更多的数据和训练步骤。有关训练过程中的回报，请参见图 3-10。

在图3-10中展示了本文所提出的模型在 MuJoCo 系列任务中的训练曲线。其中浅色部分时是模型在训练和探索时取得的分数，而实线部分是模型在训练过程中在测试环

境中测试 100 次取得的分数的平均值。由于在训练的过程中会随机选择动作以及添加噪声进行探索，因此它的分数的方差是比较大的，而在测试集上进行测试时，总是选择 Actor 输出概率最大的选项，因此它取得的分数往往比训练时的分数要好。

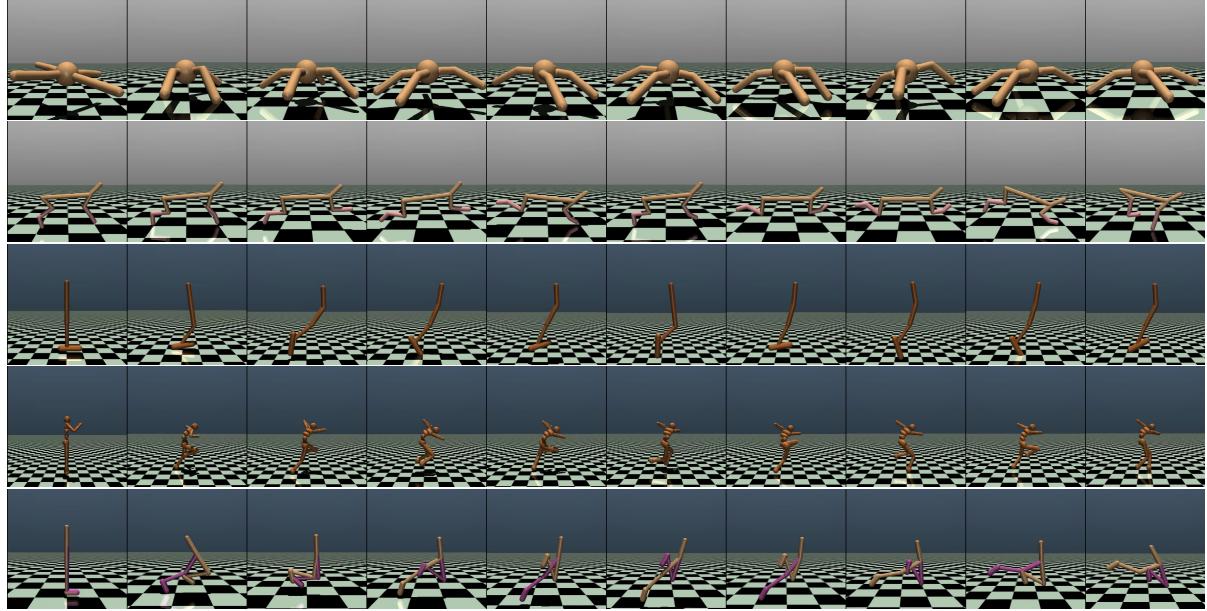


图 3-9 MuJoCo 运动姿态

3.3.4 实验细节

3.3.4.1 奖励正则化

奖励归一化在模型的训练和收敛中至关重要。原始奖励函数记作 $R_1(\mathbf{x}, A)$ ，建议将其转换为归一化形式 $R_2(\mathbf{x}, A)$ ，理想情况下应具备以下特征：

$$R_2(\mathbf{x}, A) = CR_1(\mathbf{x}, A), \quad C > 0, \quad \sup_{\mathbf{x}, A} R_2(\mathbf{x}, A) \leq 1 \quad (3-24)$$

如果可以确定一个常数 C ，通常表示为 $\frac{1}{\sup_{\mathbf{x}, A} R_1(\mathbf{x}, A)}$ ，则有以下等式成立：

$$\begin{aligned} Z_t &= R_2(\mathbf{x}_t, A_t) + \gamma R_2(\mathbf{x}_{t-1}, A_{t-1}) + \gamma^2 R_2(\mathbf{x}_{t-2}, A_{t-2}) + \dots \\ &\leq 1 + \gamma 1 + \gamma^2 1 + \dots \leq \frac{1}{1 - \gamma} \end{aligned} \quad (3-25)$$

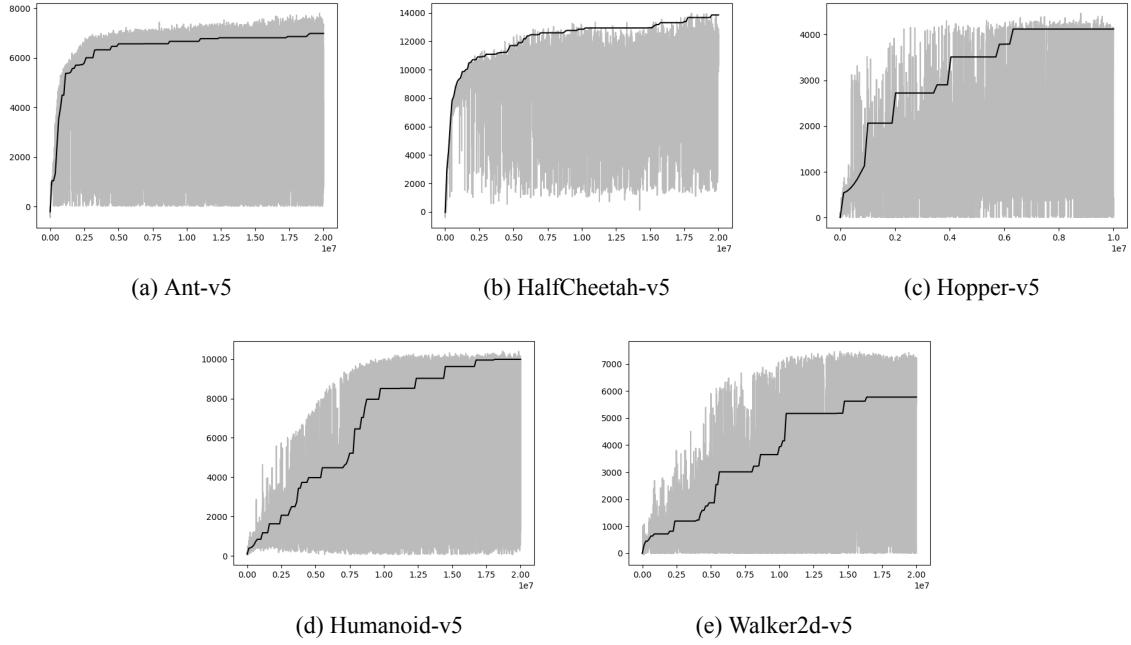


图 3-10 MuJoCo 训练曲线

考虑到上述提到的上界，建议配置超参数 $V_{MAX} = \frac{1}{1-\gamma}$ 。

3.3.4.2 对数运算

如果直接采用对数运算来计算损失函数，会导致大量的精度损失，尤其是对非常小的数值进行计算时。因此直接采用对数算子是不明智的，需要在纸面上做一些变换来规避掉这些精度损失。在实验中，部分计算使用了名为“log sum exp”的技巧。

$$\log\left(\sum_{1 \leq i \leq N} e^{x_i}\right) = x^* + \log\left(\sum_{1 \leq i \leq N} e^{x_i - x^*}\right), \quad x^* = \max_{1 \leq i \leq N} x_i \quad (3-26)$$

通过上面的变换，可以保证对数运算内的数值大于 1，从而可以避免当数值很小时对数运算会损失大量精度的问题。基于上面的讨论，“log softmax”可以表示为：

$$\log\left(\frac{e^{x_j}}{\sum_{1 \leq i \leq N} e^{x_i}}\right) = x_j - x^* - \log\left(\sum_{1 \leq i \leq N} e^{x_i - x^*}\right), \quad x^* = \max_{1 \leq i \leq N} x_i \quad (3-27)$$

更进一步地，对于累积分布地对数运算，可以表示为

$$\begin{aligned}
\log(1 - \frac{\sum_{1 \leq i \leq K} e^{x_i}}{\sum_{1 \leq i \leq N} e^{x_i}}) &= \log(\frac{\sum_{K < i \leq N} e^{x_i}}{\sum_{1 \leq i \leq N} e^{x_i}}) \\
&= (x^{**} + \log(\sum_{K < i \leq N} e^{x_i - x^{**}})) - (x^* + \log(\sum_{1 \leq i \leq N} e^{x_i - x^*})) \quad (3-28) \\
x^* &= \max_{1 \leq i \leq N} x_i, \quad x^{**} = \max_{K < i \leq N} x_i
\end{aligned}$$

在实践中，发现在构建 Policy Loss 的时候为所有的累积概率设置一个接近 0 下界（如 $1e-4$ ）会更加的鲁棒，这样可以避免 Actor 网络为了追求一些微小噪声的变化而大幅度改变策略。可以在上面的式子中添加一个乘数，来保证对数内的数值总是大于 0。

$$\begin{aligned}
&\log(1 - (1 - \epsilon) \frac{\sum_{1 \leq i \leq K} e^{x_i}}{\sum_{1 \leq i \leq N} e^{x_i}}) \\
&= \log(\frac{\epsilon \sum_{1 \leq i \leq K} e^{x_i} + \sum_{K < i \leq N} e^{x_i}}{\sum_{1 \leq i \leq N} e^{x_i}}) \\
&= \log(\frac{\sum_{1 \leq i \leq K} e^{x_i + \log(\epsilon)} + \sum_{K < i \leq N} e^{x_i}}{\sum_{1 \leq i \leq N} e^{x_i}}) \quad (3-29) \\
&= x^{**} + \log(\sum_{1 \leq i \leq K} e^{x_i + \log(\epsilon) - x^{**}} + \sum_{K < i \leq N} e^{x_i - x^{**}}) - (x^* + \log(\sum_{1 \leq i \leq N} e^{x_i - x^*})) \\
x^* &= \max_{1 \leq i \leq N} x_i, \quad x^{**} = \max(\max_{1 \leq i \leq K} x_i + \log(\epsilon), \max_{K < i \leq N} x_i)
\end{aligned}$$

3.4 总结和讨论

在本章中，提出了一种基于离散化 Actor 和 Critic 的 off-policy 深度强化学习模型，专注于机器人越障任务的应用。通过用离散分布来近似连续值和动作，该模型能够有效捕捉动作和价值空间中的多峰分布特性，并在 BipedalWalkerHardcore-v3 任务中达到了最先进的性能（见^[48] 排行榜）。此外，该模型在各种任务中的表现也接近于 TQC、TD3 和 SAC 等基准模型。

本模型和实验验证了离散化强化学习方法在机器人越障任务中的潜力，特别是在连续环境中进行动作选择的能力。机器人越障任务通常涉及复杂的地形、动态障碍物和高风险高回报的决策，这对传统强化学习算法提出了严峻挑战。尽管动作分辨率有所降

低，本文所提出的模型通过对多峰动作分布的精准学习，以及价值分布强化学习风格的 Critic 设计，成功解决了这些任务中的关键问题。

在越障任务中，例如 BipedalWalkerHardcore-v3，模型表现出了对高风险高回报场景的独特处理能力。与 TD3 和 SAC 不同，本文模型能够通过正确建模回报的分布来应对类似“爬过一个箱子”这样的高风险动作。TD3 和 SAC 倾向于建模标量回报期望，可能会错误地估计跨越障碍的动作效果，从而导致无法找到合适的策略。本模型通过离散价值分布的设计，避免了这一问题，使得机器人能够在复杂环境中更高效地规划动作。

另一个显著优势在于，本文模型通过离散化动作空间的探索机制实现了更自然、更类人的越障行为。例如，当机器人遇到障碍物时，它能够在多种策略之间进行权衡，例如跳跃、攀爬或调整步幅，而不是局限于简单的线性控制策略。这种离散化探索使得模型能够接近多峰分布的真实情况，避免了连续模型（如 TQC）中因动作分布受限而导致的不稳定步态。例如，当机器人需要跨越一个大树桩时，离散模型更倾向于生成明确的决策（如“跳跃”或“稳定平衡”），而连续模型可能在中间策略中失败，导致机器人摔倒。

在步态控制方面，本模型还展示了较高的适应性和鲁棒性。在平坦地面上，模型学会了节能的交替步态；在上下坡时，能够自动调整步幅和节奏；在越过小障碍物时表现出精确的步幅控制；而面对大型障碍物时，则通过攀爬动作完成任务。这些特性使得机器人不仅能够应对复杂的越障任务，还能够在不同地形和环境中动态切换步态，避免了重复学习的成本。

作为离散化强化学习在机器人越障任务中的初步尝试，本章模型仍有改进空间。一个潜在的改进方向是对离散化动作进行插值处理。例如，可以通过核函数将多个离散化动作映射到一个连续动作，以提高动作分辨率，从而进一步提升性能。尽管如此，实验结果表明，本模型已在环境复杂的越障任务中展现出良好的适应性和鲁棒性，具备实际应用的潜力。

总之，本研究表明，将离散化动作空间应用于复杂越障任务是可行的，并且在现实环境中具有广阔的应用前景。这一方法有望成为未来机器人步态控制与越障任务的有效工具。

第4章 布尔网络与序列建模

4.1 背景

在避障和越障任务中，机器人通常还需要依赖环境的历史信息来校准和同步自身的任务规划，从而做出精准的决策，这使得序列建模显得尤为重要。例如，在记忆力导航任务中，机器人需要记住先前探索的路径、遇到的障碍物位置，以及可能的通行路线。这类任务不仅需要对当前环境状态进行感知，还要求对过往的状态进行有效的存储和处理，以避免重复探索错误的路径，或者在复杂的地形中选择最优的越障策略。这种需求超出了马尔可夫决策过程框架中的瞬时状态转移，要求模型具备强大的序列建模能力来提取和利用历史信息，从而实现更高效的自主任务规划与动态调整。

序列建模有着非常广泛的应用前景。对于 Transformer 模型而言，它确实可以根据 Prompt 来匹配到需要的信息，但这并不代表它对于信息具有强的加工处理的能力。可以来做个简单的实验，让 Transformer 来学习输出 01 序列内有多少个 1。例如，对于 000101，希望 Transformer 输出 2；00101001，输出 3，等等。实际上 Transformer 会表现得非常糟糕，因为该任务要求模型不仅对内容进行匹配，更需要对内容进行思考和加工。

强化学习基于马尔可夫决策过程的框架，实际上规避了对于复杂上下文的表征过程。而表征，是逻辑推理的关键，好的表征可以让推理事半功倍。机器学习的核心也应该是表征。实际上，代数表示论对于理解机器学习中的表征有非常重要的参考意义。

代数表示论主要研究代数结构（如群、环、代数等）与线性空间之间的关系。具体来说，它关注如何通过线性变换来“表示”代数对象。代数表示论在表示一个有限群时，会将群内的每一个元素映射到矩阵空间内，使得这些矩阵以及矩阵的乘法构成的代数结构与原来的群同构。

布尔网络相较于主流的深度学习框架来说更具可解释性，更适合用于逻辑推理。希望通过可以通过序列数据来构建布尔网络模型，使得布尔网络可以解释和预测系统的行为。辛普森悖论（Simpson's paradox）揭示了概率论框架下做因果推断的局限性，因此认为

布尔网络是一个值得尝试的方向。

本节工作尝试离开深度学习和概率论的主流框架，探寻一种有利于序列建模、逻辑推理的机器学习方法，观察记忆力形成的基本原理。提出了布尔算子，该算子通过重新定义加法和乘法，再结合矩阵乘法运算的规则，可建模本节所列举三个记忆任务，为构建基于记忆力的机器学习模型提供思路。

4.2 模型与方法

4.2.1 序列建模

考虑一个有限的序列数据集，

$$\begin{aligned} u_0^0, u_1^0, \dots, u_{t_0}^0 &\rightarrow y^0 \\ u_0^1, u_1^1, \dots, u_{t_1}^1 &\rightarrow y^1 \\ u_0^2, u_1^2, \dots, u_{t_2}^2 &\rightarrow y^2 \\ &\vdots \\ u_0^N, u_1^N, \dots, u_{t_N}^N &\rightarrow y^N \end{aligned}$$

每一个数据集内有限长度的 u 序列都对应于可以观察到的 y 标签。希望构建出一个布尔状态空间方程来建模这一数据集。

4.2.2 状态空间方程

状态空间方程是描述系统规律的常用手段。一个离散状态空间方程可以表示为：

$$x_t = f(x_{t-1}, u_{t-1}) \quad (4-1)$$

$$y_t = g(x_t) \quad (4-2)$$

其中， x 是状态变量， y 是可观察变量， u 是外部控制变量。

关心的是非线性的状态空间方程。为了简化问题。设置状态空间方程内每一个变量都是布尔变量，每一个函数都是由“与或非”构成的布尔函数。

布尔代数可以表示成这样一个域 $K = \{0, 1\}$ ，其中对于 $a, b \in K$

$$a + b = a \vee b \quad (4-3)$$

$$a \times b = a \wedge b \quad (4-4)$$

下面这个形式称为稀疏布尔状态空间

$$x_t = A(u_{t-1})x_{t-1} \quad (4-5)$$

$$y_t = Cx_t \quad (4-6)$$

式子中的矩阵运算为域 K 上的矩阵运算。 $A(u_{t-1})$ 是与 x 维度一致的方阵。每一个有限维的布尔状态空间方程，都与一个有限维的稀疏布尔状态空间方程同构。

困难的是如何从已知的稀疏布尔状态空间方程中得到一个紧致的布尔状态空间方程的表达。这涉及到如何对状态空间中的每一个点进行编码，使得方程的维度能够降下来，计算的成本能够达到最低。拿二进制加法来举例，表达一个 32 位的加法需要有 2^{32} 个状态，使用稀疏的方式来表达是不切实际的。对于状态数不高的稀疏编码，可以暴力遍历所有编码方案，找到最简洁紧凑的一组。

对于比较简单的问题，可以考虑从数据中学习一个同构映射 $A : U \rightarrow K_{n \times n}$ ， U 表示由 u 构成的半群， $K_{n \times n}$ 表示 K 上的 n 阶方阵空间。使得 A 满足

$$A(u_0 u_1 \cdots u_t) = A(u_0)A(u_1) \cdots A(u_t) \quad (4-7)$$

但是对于复杂的模型，矩阵的维度会变得非常的大，使得一切都不可行。或许可以

考虑给模型添加一些层级结构。这也可能是对状态空间进行编码的一种途径。

还需要考虑模型计算的并行性。对于维度不太高的网络，结合律可以提高计算过程的并行性。

4.2.3 布尔算子

可以在实数集上定义一种新的加法和乘法， $(R, +', \times')$ 。

$$a +' b = \log(e^{a+\epsilon_1} + e^{b+\epsilon_2}) \quad (4-8)$$

$$a \times' b = -\log(e^{-(a+\epsilon_1)} + e^{-(b+\epsilon_2)}) \quad (4-9)$$

其中 ϵ_1, ϵ_2 表示服从正态分布（或者其他）的有界的随机噪声。当这个只考虑正负无穷时，它与布尔运算是等价的。但如果将集合扩张到实数域 \mathbf{R} 的话，它就不满足分配律了。这样设计的目的是为了引入基于梯度的数值优化的算法。因为 $+$ 和 \times' 作为可求导的连续二元函数却与不连续的 \max 和 \min 有着非常相似的性质，所以它非常适合用来看逻辑运算。记 sigmoid 函数为 $\sigma(t) = \frac{1}{1+e^{-t}}$ ，那么就有

$$\mathbb{E}\sigma(a \times' b) \leq \min(\mathbb{E}\sigma(a), \mathbb{E}\sigma(b)) \quad (4-10)$$

$$\mathbb{E}\sigma(a +' b) \geq \max(\mathbb{E}\sigma(a), \mathbb{E}\sigma(b)) \quad (4-11)$$

\mathbb{E} 表示随机变量的期望。可以为每一个 u 在 $(R, +', \times')$ 设置一个可学习的矩阵 $A(u) = A_u$ ，尽管加法和乘法一般不满足分配律，但它们在无穷处是满足的。如果有两个布尔矩阵 A, B ，那么可以定义它的积 C 为：

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1q} \\ a_{21} & a_{22} & \cdots & a_{2q} \\ \vdots & & & \\ a_{p1} & a_{p2} & \cdots & a_{pq} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1r} \\ b_{21} & b_{22} & \cdots & b_{2r} \\ \vdots & & & \\ b_{q1} & b_{q2} & \cdots & b_{qr} \end{bmatrix}$$

$$C = A \times B = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1r} \\ c_{21} & c_{22} & \cdots & c_{2r} \\ \vdots & & & \\ c_{p1} & c_{p2} & \cdots & c_{pr} \end{bmatrix}$$

$$c_{ij} = \log\left(\frac{1}{e^{-(a_{i1} + \epsilon_{i1}^1)} + e^{-(b_{1j} + \epsilon_{1j}^2)}} + \frac{1}{e^{-(a_{i2} + \epsilon_{i2}^1)} + e^{-(b_{2j} + \epsilon_{2j}^2)}} + \cdots + \frac{1}{e^{-(a_{iq} + \epsilon_{iq}^1)} + e^{-(b_{qj} + \epsilon_{qj}^2)}}\right)$$

对于一个具体的序列而言 $u_0^k, u_1^k, \dots, u_{t_k}^k \rightarrow y^k$, 希望模型符合下面的式子。

$$x_{t_k} = \left(\prod_{i=0}^{t_k} A(u_i^k)\right) x_0 \tag{4-12}$$

$$Cx_{t_k} = y^k \tag{4-13}$$

上面式子中的向量和矩阵都是 $(R, +', \times')$ 上定义的。可以根据 Cx_{t_k} 和 y^k 的差异设置损失函数, 如交叉熵, 通过梯度下降来进行搜索。不过模型的凸性没有办法保证, 为了跳出局部最优解, 需要为每一个参数添加噪声。当模型参数收敛到无穷时, 任何有界的噪声对于模型都是没有影响的, 所以希望这些噪声可以帮助跳出来。暂时也没有其他太好的优化算法。

4.3 实验

记忆力是机器学习模型做出精确逻辑推理的最重要因素, 是处理长上下文任务基本原理, 并且也有利于增强模型的可解释性。这些任务可以帮助理解当前主流的“Trans-

former”模型主要的局限性。本节的实验包括三个基本记忆任务。

Remember

要介绍的第一个任务叫“Remember”，在 $t = 0$ 时刻，外界会随机向系统展现一张卡片 i , token “[card i]”; 在 $t = 1, \dots, n - 1$ 时刻，外界会向系统输入 token “[wait]”，意味着系统只需要等待不需要作任何事情；而在 $t = n$ 时刻，外界会对系统发出提问，输入 token “[ask]”，系统需要回复最开始看到的那张卡片是什么，要做到这一点，系统需要利用自身能力对先前接收的输入进行记忆；最后，外界再输入结束指令，token “[end]”。举例来说，它们动作序列和观测序列分别为：

t	0	1	2	\dots	n	$n + 1$	$n + 2$	\dots
y	$None$	[wait]	[wait]	[wait]	[wait]	[cardi]	[end]	[end]
u	[cardi]	[wait]	[wait]	[wait]	[ask]	[end]	[end]	[end]

Count

在 $t = 0, \dots, n - 1$ 时刻，外界会随机输入一些 0 和 1；而在 $t = n$ 时刻，外界会对系统发出提问，输入 token “[ask]”，系统需要回复之前总共输入了多少个 1, 加起来三多少；最后，外界再输入结束指令，token “[end]”。

t	0	1	2	\dots	n	$n + 1$	$n + 2$	\dots
y	$None$	[wait]	[wait]	[wait]	[wait]	$\sum_{i=0}^{n-1} u_i$	[end]	[end]
u	0	0	1	0 or 1	[ask]	[end]	[end]	[end]

Repeat

该任务考察的是短期记忆能力，外界会不断向系统随机展示不同的卡片，token “[card i_t]”；在 $t = n$ 时刻，外界会对系统发出提问，输入 token “[ask]”，系统需要返回 $t = n - m$ 时刻出现了什么卡片。

t	0	1	2	\dots	n	$n + 1$	$n + 2$	\dots
y	$None$	[wait]	[wait]	[wait]	[wait]	[cardi _{$n-m$}]	[end]	[end]
u	[cardi ₀]	[cardi ₁]	[cardi ₂]	[cardi _{t}]	[ask]	[end]	[end]	[end]

这三个任务参考自 [https://github.com/proroklab/popgym^{\[50\]}](https://github.com/proroklab/popgym)。不过对其做了一些简化处理，使得可以更专注于核心的问题。

这些纯粹的符号推理任务与自然语言的一些任务相比也有显著的差别，往往符号推理任务不会找到非常固定的模式来对结果进行预测，这与遵循一定语法结构的自然语言有着明显的区别。符号推理的任务注重的是因果推断的能力。

对于 Remember 而言，它的状态数量与卡片数量 c 相关。对于 Count 来说，状态数量与最大的求和数相关。对于 Repeat 来说，它的状态数是 c^m 。总体而言，这三个任务的状态数是依次递增的，在不同的数量级。

Table 4-1列举了实验的结果。结果表明，布尔模型可以解决这里所列举的三个任务。而 Transformer 模型在 Remember 和 Repeat 两项任务中有效，而在 Count 任务中却无法取得效果。Remember 和 Repeat 任务都是根据需要从序列中检索出信息，不需要对信息进行额外的加工；而 Count 任务由于需要对序列进行加法计数，仅仅依赖与检索能力是难以做到的。这样的实验结果很好地反映了布尔模型与 Transformer 在结构上的差异，Transformer 模型依赖于自身的注意力机制能够起到很好的检索效果，但也限制了它在推理方面的能力。相关的代码公开在 <https://github.com/ZJEast/Semigroup-Learning>

Task	Transformer	布尔
Remember	有效	有效
Count	无效	有效
Repeat	有效	有效

表 4-1 Transformer 与布尔模型在三个任务上有效性的比较

4.4 总结与讨论

假设有一个很长的 01 序列，要求模型数出里面有多少个 1（结果可能是一个很大的数），应该怎么做？或者换个说法，如何通过数据来构建一个加法计数器？一个基本的想法是引入转译层的概念，所谓转译，就是让当前的层学习去输出下一层需要的 token，拿下面这个序列举例：

<i>layer1 :</i>	0	1	0	1	1	1	1	1	0	1	0	0	1	1	0	1	0	1	1	0	<i>feature1 = 0</i>
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
<i>layer2 :</i>	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	<i>feature2 = 0</i>
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
<i>layer3 :</i>	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	<i>feature3 = 1</i>
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
<i>layer4 :</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	<i>feature4 = 1</i>

通过多层转译，可以使用各层的特征来一起对系统的状态空间进行表征。就这一个加法的例子而言，如果原来的系统是 2^n 维度的话，可以把它拆成 n 个维度为 2 的系统。当然这个例子可能过于简单，但这个思路对于更一般的一些问题可能是有启发性的。这相当于在研究群论时，会尝试分离出子群和商群，从一团乱麻的关系中分离出几个相对独立的几个部分，利用这些相对独立的性质来进行高效的表征。还可以在一层之中设置多个通道的特征，考虑不同通道之间的融合。

通过本工作的实验和分析，可以看出在序列建模和逻辑推理的结合上，仍有以下方向值得进一步探讨：(1) 结合强化学习和博弈论。当完全信息不可直接获取时，使用布尔网络来对状态进行表征和推导。利用强化学习来让模型学习使用外部储存工具，构建思维链，为具身只能提供基础。(2) 找到一套分析方法来刻画一个序列数据集的可被学习的难易程度。需要回答，到底什么样的问题是不可以被认知和学习的，会不会存在一些问题是难以系统的方式建模的？(3) 降低系统的维度是一个非常棘手的问题，一方面是系统状态空间中点的数量，另一方面是需要找到一种合理的编码方式，使得能够体现不同状态点之间的关联，使得在计算状态转移时不需要付出过高的计算成本。或许这些问题也有一些没被发现的数学的规律。(4) 从概率论的角度审视人工智能在某些任务上可能存在一定的局限性，尤其是符号推理任务。在这种情况下，非黑即白的演绎推理或许更有助于揭示事物之间的因果关系，同时还能更好地利用反事实来提升模型的能力。(5) 布尔网络目前来说还是处理序列问题比较自然，或许未来可以考虑应用到视觉领域。(6) 机器学习与传统的数学体系展现出非常多不一样的特点。传统的数学体系把重点放在分析上面，也就是在已知的前提下推导结论。而机器学习却不然，它是从事后

的结论中反过来构造出一个前提，往往这个前提并不唯一。分析和构造都是重要的方法论，都是认识自然、追求真理的有力工具。

这是一次结合符号主义和连接主义的尝试。这一过程也尝试丢弃许多深度学习才有的概念，所设计的网络也不存在梯度爆炸的问题，没有各种 Normalization，没有序列长度的限制，没有模型深度的限制。参考了一些循环神经网络方面的工作^[51,52]。控制论和表示论的一些概念可以针对人工智能问题提供一些新的视角。通过学习每一个单词在矩阵空间中的表示，可以实现符号推理的效果。可解释性和可控性可能会更好一些。

在未来的机器人自主任务规划中，尤其是面对复杂的避障或越障任务时，如走迷宫或跨越不规则地形，智能体不仅需要实时感知当前的环境状态，还必须利用历史序列数据来校准和同步自身的任务规划。这要求机器人具备一定的记忆能力，以便从过去的经验中提取关键信息，优化当前的决策和行为路径。因此，序列分析学习成为提升任务执行能力的核心技术之一。

在进行序列分析学习时，布尔网络作为一种能够有效建模逻辑推理和因果关系的框架，展现出极大的潜力。与传统的强化学习模型相比，布尔网络能够为智能体提供更高层次的推理和记忆能力，尤其是在面对复杂的多步决策任务时。在这种情境下，布尔网络不仅能够帮助机器人理解任务中不同阶段的关系，还能够预测在特定环境条件下可能出现的多种情境，进而优化任务执行。

更为重要的是，布尔网络与强化学习的结合，为解决需要记忆力和长期依赖关系的问题提供了新思路。在强化学习框架下，智能体通常面临稀疏奖励和动态环境的挑战，布尔网络能够通过对历史数据的编码和推理，使得智能体能够更有效地处理和学习这些复杂的任务。未来，随着布尔网络与强化学习的深入结合，机器人将在自主任务规划中实现更高效的决策支持，提升在动态、复杂环境中的适应能力和执行效率。

第 5 章 结语

本文研究聚焦于机器人避障与越障的强化学习方法，以应对复杂环境中的多样化任务需求，提出了一系列具有创新性的技术路径。避障和越障是机器人执行任务时的关键技能，其核心挑战在于如何在动态环境中实现灵活避障、精确越障以及应对长期依赖的序列任务。为此，本文通过系统的算法设计，分别在避障策略优化、越障任务建模以及序列分析学习方面取得了新的进展，并展示了布尔网络与强化学习结合的潜力。

在避障任务中，本文提出了一种目标驱动的稀疏奖励强化学习框架，将障碍物规避与目标导向任务无缝结合。通过后见经验回放（HER）和基于目标与障碍物相对位置的特征建模，智能体能够实时适应环境变化，动态调整决策过程。这种方法避免了对精确物理模型的依赖，提升了算法在动态场景中的鲁棒性和灵活性。实验结果表明，该方法在无人驾驶和机器人导航等领域展现出重要的实际价值。

针对越障任务，本文引入了一种基于离散化 Actor-Critic 模型的创新方法，将连续动作空间离散化以支持多峰策略分布，同时结合离散价值分布建模任务中的不确定性。这种设计克服了传统连续模型在复杂任务中的单一峰值陷阱问题，显著提高了智能体在复杂环境中的策略多样性和决策稳定性。尤其在具有高动态和多峰任务需求的场景中，该模型展现了卓越的表现，为复杂的越障任务提供了一种可靠的解决方案。

此外，针对复杂任务中的长期依赖关系，本文探讨了布尔网络与序列分析学习的结合。这一方法在自主任务规划中嵌入了逻辑推断能力，使得智能体能够在历史数据中提取关键信息，校准自身的任务规划与执行。这种结构对需要路径记忆与多步推断的任务尤其有效，显著提升了智能体在长时间依赖环境中的适应性。尽管布尔网络在状态空间增长时可能带来计算复杂性，本文的研究初步验证了其在序列分析学习中的潜力，为后续优化和大规模应用提供了方向。

综上所述，本文的研究从避障、越障和序列建模三个方面，为机器人强化学习在复杂任务中的应用提供了新的解决方案。避障方法强调了实时环境适应能力，越障模型提升了复杂决策任务中的策略多样性，而布尔网络与序列分析的结合则为长时推断问题带来了全新的视角。尤其在自主任务规划和序列分析学习中，布尔网络的前景值得进一步

探索，以应对未来更加复杂的任务需求。这些工作不仅扩展了强化学习技术在机器人任务中的适用范围，也为实际应用中的复杂场景提供了重要的理论支撑和实践参考。

参考文献

- [1] BELLEMARE M G, DABNEY W, MUNOS R. A distributional perspective on reinforcement learning[C] // International conference on machine learning. 2017 : 449–458.
- [2] MNIH V, OTHERS. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518 : 529–533.
- [3] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. [S.l.] : MIT press, 2018.
- [4] BOTVINICK M, OTHERS. Reinforcement learning, fast and slow[J]. Trends in Cognitive Sciences, 2019, 23 : 408–422.
- [5] SILVER D, OTHERS. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529 : 484–489.
- [6] LAKE B M, OTHERS. Building machines that learn and think like people[J]. Behavioral and Brain Sciences, 2017, 40.
- [7] BROWN T, OTHERS. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020.
- [8] LEVINE S, OTHERS. End-to-end training of deep visuomotor policies[J]. Journal of Machine Learning Research, 2016, 17(1) : 1334–1373.
- [9] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C] // International conference on machine learning. 2018 : 1861–1870.
- [10] LILLICRAP T P, OTHERS. Continuous control with deep reinforcement learning[J]. ICLR, 2016.
- [11] DULAC-ARNOLD G, OTHERS. Deep reinforcement learning in large discrete action spaces[J]. arXiv preprint arXiv:1512.07679, 2015.
- [12] TANG H, OTHERS. Exploration: A study of count-based exploration for deep reinforcement learning[J]. Advances in Neural Information Processing Systems, 2017.

- [13] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE transactions on Systems Science and Cybernetics, 1968, 4(2) : 100 – 107.
- [14] KUFFNER J J, LAVALLE S M. RRT-connect: An efficient approach to single-query path planning[C] // Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065) : Vol 2. 2000 : 995 – 1001.
- [15] RAIBERT M H. Hopping in legged systems—Modeling and simulation for the two-dimensional one-legged case[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1984, 14(3) : 451 – 463.
- [16] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.
- [17] FUJIMOTO S, HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[C] // International conference on machine learning. 2018 : 1587 – 1596.
- [18] ANDRYCHOWICZ M, WOLSKI F, RAY A, et al. Hindsight experience replay[J]. Advances in neural information processing systems, 2017, 30.
- [19] KUZNETSOV A, SHVECHIKOV P, GRISHIN A, et al. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics[C] // International Conference on Machine Learning. 2020 : 5556 – 5566.
- [20] DABNEY W, ROWLAND M, BELLEMARE M, et al. Distributional reinforcement learning with quantile regression[C] // Proceedings of the AAAI Conference on Artificial Intelligence : Vol 32. 2018.
- [21] TOWERS M, TERRY J K, KWIATKOWSKI A, et al. Gymnasium[EB/OL]. Zenodo, 2023.
<https://zenodo.org/record/8127025>.
- [22] RAFFIN A, HILL A, GLEAVE A, et al. Stable-Baselines3: Reliable Reinforcement Learning Implementations[J/OL]. Journal of Machine Learning Research, 2021, 22(268) : 1 – 8.
<http://jmlr.org/papers/v22/20-1364.html>.
- [23] DABNEY W, OSTROVSKI G, SILVER D, et al. Implicit quantile networks for distributional reinforcement learning[C] // International conference on machine learning. 2018 : 1096 – 1105.
- [24] VASERSTEIN L N. Markov processes over denumerable products of spaces, describing large systems of automata[J]. Problemy Peredachi Informatsii, 1969, 5(3) : 64 – 72.

- [25] BARTH-MARON G, HOFFMAN M W, BUDDEN D, et al. Distributed distributional deterministic policy gradients[J]. arXiv preprint arXiv:1804.08617, 2018.
- [26] CHRISTODOULOU P. Soft actor-critic for discrete action settings[J]. arXiv preprint arXiv:1910.07207, 2019.
- [27] NEUNERT M, ABDOLMALEKI A, WULFMEIER M, et al. Continuous-Discrete Reinforcement Learning for Hybrid Control in Robotics[J/OL], 2020. <https://arxiv.org/abs/2001.00449>.
- [28] METZ L, IBARZ J, JAITLEY N, et al. Discrete Sequential Prediction of Continuous Actions for Deep RL[J/OL], 2019. <https://arxiv.org/abs/1705.05035>.
- [29] TANG Y, AGRAWAL S. Discretizing Continuous Action Space for On-Policy Optimization[J/OL], 2020. <https://arxiv.org/abs/1901.10500>.
- [30] LUO J, DONG P, WU J, et al. Action-quantized offline reinforcement learning for robotic skill learning[C] // Conference on Robot Learning. 2023 : 1348 – 1361.
- [31] FAREBROTHER J, ORBAY J, VUONG Q, et al. Stop Regressing: Training Value Functions via Classification for Scalable Deep RL[J/OL], 2024. <https://arxiv.org/abs/2403.03950>.
- [32] FOX D, BURGARD W, THRUN S. The dynamic window approach to collision avoidance[J]. IEEE Robotics & Automation Magazine, 1997, 4(1) : 23 – 33.
- [33] LAVALLE S M. Rapidly-exploring random trees: A new tool for path planning[J]. Technical Report, 1998.
- [34] FALCONE P, BORRELLI F, ASGARI J, et al. Predictive active steering control for autonomous vehicle systems[J]. IEEE Transactions on Control Systems Technology, 2007, 15(3) : 566 – 580.
- [35] PATHAK D, AGRAWAL P, EFROS A A, et al. Curiosity-driven exploration by self-supervised prediction[C] // International conference on machine learning. 2017 : 2778 – 2787.
- [36] METELLI A M, MUTTI M, RESTELLI M. Configurable Markov decision processes[C] // International Conference on Machine Learning. 2018 : 3491 – 3500.
- [37] SCHAUL T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay[J]. arXiv preprint arXiv:1511.05952, 2015.

- [38] SAKAI A, INGRAM D, DINIUS J, et al. Pythonrobotics: a python code collection of robotics algorithms[J]. arXiv preprint arXiv:1808.10703, 2018.
- [39] VEZHNEVETS A S, OSINDERO S, SCHAUER T, et al. FeUDal Networks for Hierarchical Reinforcement Learning[C/OL] // PRECUP D, TEH Y W. Proceedings of Machine Learning Research, Vol 70 : Proceedings of the 34th International Conference on Machine Learning. [S.l.] : PMLR, 2017 : 3540 – 3549. <https://proceedings.mlr.press/v70/vezhnevets17a.html>.
- [40] HUANG S, DOSSA R F J, YE C, et al. CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms[J/OL]. Journal of Machine Learning Research, 2022, 23(274) : 1 – 18. <http://jmlr.org/papers/v23/21-1342.html>.
- [41] BADUE C, GUIDOLINI R, CARNEIRO R V, et al. Self-driving cars: A survey[J]. Expert Systems with Applications, 2021, 165 : 113816.
- [42] KOBER J, BAGNELL J A, PETERS J. Reinforcement Learning in Robotics: A Survey[J/OL]. Springer Tracts in Advanced Robotics, 2014, 97 : 9 – 67. <http://dx.doi.org/10.1007/978-3-319-03194-1-2>.
- [43] LILlicrap T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[C] // 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings. 2016.
- [44] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.
- [45] YU Z, MOHAMMED A, PANAHİ I. Review of three PWM techniques[J/OL]. Proceedings of the American Control Conference, 1997, 1(June) : 257 – 261. <http://dx.doi.org/10.1109/acc.1997.611797>.
- [46] HUGHES A, DRURY B. Induction Motors – Rotating Field, Slip and Torque[G/OL] // Electric Motors and Drives. [S.l.] : Elsevier, 2013 : 141 – 167. <http://linkinghub.elsevier.com/retrieve/pii/B978008098332500005X>. <https://linkinghub.elsevier.com/retrieve/pii/B978008098332500005X>.
- [47] ESTEKI M, KHAJEHODDIN S A, SAFAEE A, et al. LED Systems Applications and LED Driver Topologies: A Review[J/OL]. IEEE Access, 2023, 11(April) : 38324 – 38358. <http://dx.doi.org/10.1109/ACCESS.2023.3267673>.

- [48] OPENAI. Leaderboard[EB/OL]. . <https://github.com/openai/gym/wiki/Leaderboard#bipedalwalker-v2-and-bipedalwalker-v3>.
- [49] WEI H, YING L. FORK: A Forward-Looking Actor For Model-Free Reinforcement Learning[J], 2021.
- [50] MORAD S, KORTVELESY R, BETTINI M, et al. POPGym: Benchmarking Partially Observable Reinforcement Learning[C/OL] // The Eleventh International Conference on Learning Representations. 2023. <https://openreview.net/forum?id=chDrutUTs0K>.
- [51] PENG B, ALCAIDE E, ANTHONY Q, et al. Rwkv: Reinventing rnns for the transformer era[J]. arXiv preprint arXiv:2305.13048, 2023.
- [52] GU A, DAO T. Mamba: Linear-time sequence modeling with selective state spaces[J]. arXiv preprint arXiv:2312.00752, 2023.

攻读硕士学位期间相关的科研成果目录

1. 发明专利

(1) 基于强化学习的车辆局部避障算法及系统[P]. 中国发明专利. CN 118605523

A. 2024.09.06

(2) 一种基于强化学习的机器人控制方法及系统[P]. 中国发明专利. CN 117850241

A. 2024.04.09

致谢

由衷感谢我的导师魏天琪副教授，本文是在他的指导下完成的。强化学习是一个有挑战的领域，我时常对遇到的困难感到沮丧和懊恼。感谢魏老师对我的包容，支持我做自己感兴趣的领域。

张俊东

2024年11月25日

