分类号：TN92　　单位代码：10335

密　级：公开　　学　　号：21960197

浙江大学

硕士学位论文

中文论文题目：基于多智能体强化学习的车联网频谱接入方法研究

英文论文题目：Multi-agent Reinforcement Learning based Spectrum Access Mechanism Design for Vehicular Networks

申请人姓名：向平

指导教师：单杭冠　副教授

专业学位类别：工程硕士

专业学位领域：电子与通信工程

所在学院：信息与电子工程学院

论文提交日期　2022年1月

基于多智能体强化学习的车联网频谱接入方法研究

论文作者签名：向平

指导教师签名：单杭冠

论文评阅人1：匿名

评阅人2：匿名

评阅人3：匿名

评阅人4：

评阅人5：

答辩委员会主席：赵民建/教授/浙江大学信电学院

委员1：郑史烈/教授/浙江大学信电学院

委员2：余官定/教授/浙江大学信电学院

委员3：林宏焘/高级/浙江大学信电学院

委员4：何先华/高级工程师/诺基亚通信投资（中国）有限公司

委员5：

答辩日期：2022年3月10日

浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得浙江大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：向平　　签字日期：2022年3月11日

学位论文版权使用授权书

本学位论文作者完全了解浙江大学有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权浙江大学可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：向平　　导师签名：单祝冠

签字日期：2022年3月11日　　签字日期：2022年3月11日

致　　谢

时光荏苒，三年硕士生活转瞬即逝，回首入学之际仿佛就在昨天。回顾过去三年的求学之路，虽然经历了诸多挫折坎坷，但自己终究是坚持了过来，这段经历将使我受益终生。在此，谨向我的导师、实验室伙伴、朋友及家人致以最真挚的感谢与最美好的祝愿。

首先，我想感谢我的导师单杭冠老师。在学术科研方面，单老师以其严谨细致的态度、开阔前沿的思想、勤奋努力的作风为我树立了良好的榜样。单老师时常教导我要做出有价值的工作，在短暂的三年硕士生活期间，我虽然努力朝着这个方向前进，但自觉仍存在巨大的不足。单老师在科研方面的指导将是我一生的财富，值得我在未来持续践行之。单老师在日常生活里也给予了我很多关怀，让我能够不受外界纷扰，专心科研。在此，我向单老师这三年以来的付出表示感恩!

然后，我要感谢实验室的诸多伙伴。感谢已经毕业的袁建涛博士、何泓利博士、甄晓健硕士、何中杰硕士、洪春华硕士，杨鲲鹏硕士、郑琪铭硕士、潘景硕士，各位师兄师姐作为我科研道路上的前辈，给予了我很多无私的帮助。此外，我还要感谢Ameer博士、宋美燕博士、张越博士、李娜博士、岳之凌博士、程琦硕士、谢志文硕士、梁昊羿硕士、鄢嗣芾硕士，各位共同为实验室营造了积极乐观的氛围，与你们的相处让我受益颇多。

感谢所有帮助过我的朋友，不管是实际生活中的好友，还是虚拟世界中的网友，感谢你们温暖了我枯寂的生活。

最后，我最诚挚的感谢要献给我的父母和家人们，是你们在我人生的前二十四年里为我承担了所有，为我创造了最好的生活和学习条件，在我迷茫、疲惫的时候，是你们的问候与嘱咐为我指明了方向，你们永远是我最温暖最踏实的庇护所。感谢你们持续的付出，是你们，为我树立了人生的榜样。

在今后的日子里，我将以更加积极、乐观、坚强的态度去迎接未知的挑战。不忘初心，砥砺前行!

向平

2021年12月

摘　要

车联网（Vehicle-to-everything, V2X）通信将是未来智慧道路交通体系中不可或缺的一环，V2X技术将道路交通中的各参与主体联系起来，能够提升道路交通效率、减少交通事故、增强安全、丰富驾驶者车载体验。由于车联网通信系统中可能同时存在多种类型用户设备（Vehicle User Equipment, VUE），其服务质量（Quality-of-service, QoS）需求各不相同，为了满足不同业务的QoS需求，需要对频谱资源进行合理分配。然而，车联网通信中，由于频谱资源较为稀缺，以及车辆高速移动导致信道具有快速时变特性，限制了全局信息获取的可行性，使得传统基于中心式优化的设计思路具有一定的局限性。因此，为了适应动态、复杂的车联网环境，设计分布式的车联网频谱接入机制有重要意义。与此同时，人工智能（Artificial Intelligence, AI）技术，如深度强化学习（Deep Reinforcement Learning, DRL），开辟了数据驱动的全新算法设计思路，近年来在无线通信领域中的应用越来越广泛。本文研究了蜂窝车联网（Cellular V2X, C-V2X）系统中V2I（Vehicle-to-infrastructure）用户和V2V（Vehicle-to-vehicle）用户共存场景下的分布式频谱接入机制设计。基于多智能体强化学习（Multi-agent Reinforcement Learning, MARL），本文以最大化V2I用户总吞吐量同时满足V2V用户时延可靠性需求为优化目标，提出了两种分布式的频谱接入算法。

首先，通过将车联网频谱接入优化问题建模为分布式部分可观测马尔可夫决策过程（Decentralized Partially Observable Markov Decision Process, Dec-POMDP），本文提出了一种基于MARL的智能体间完全独立工作的分布式频谱接入算法。在该算法中，智能体使用不包含信道状态信息（Channel State Information, CSI）的局部环境观测作为输入，来联合优化子信道和传输功率的选择。通过奖励函数和训练机制的设计，多个智能体学习形成隐式的协作模式。为了更好地适应车联网环境，提升学习性能，该算法集成了一系列先进的DRL技术，并且为了应对多智能体并发学习所引起的非平稳性，该算法还引入了滞后学习机制和并发经验回放（Concurrent Experience Replay Trajectories, CERT）机制，以稳定训练过程。为了解决车联网环境动态变化导致智能体难以训练的问题，该算法还引入了一种近似遗憾奖励（Approximate Regretted Reward, ARR）机制来实现更准确的训练效果评估。仿真结果验证了该算法在V2I用户总吞吐量和V2V用户包交付率两个指标上相对于对比方

案的性能优势，并表明所提算法具有良好的稳健性和扩展性。

为了实现更好的协作效果，本文还将通信机制引入MARL。考虑智能体不仅学习有效的信道接入策略，并且还通过学习形成通信协议，本文提出了一种通过结合智能体间通信机制来实现显式协作的分布式频谱接入控制算法。在该算法中，智能体由动作选择模块和消息产生模块组成：动作选择模块选择频谱接入动作，而消息产生模块则负责生成交互信息，两个模块均由深度神经网络（Deep Neural Network, DNN）实现，且共享网络参数。为了实现消息产生模块的端到端训练，在算法设计中引入了离散/正则处理单元（Discretize/regularize Unit, DRU），其在训练阶段对消息产生模块的输出正则化，而在测试阶段离散化，使得消息产生模块的参数可以通过损失函数进行梯度回传更新，实现端到端训练。为减小训练开销，所有智能体共享网络参数，由于每个智能体对环境的观测结果各不相同，因此智能体能表现出不同的行为策略。最终，仿真结果验证了引入通信机制的有效性，所提算法能够实现比同样基于MARL的完全独立工作、没有通信机制的算法更优的性能，表明通信交互机制能够有效提升智能体间的协作效果。

关键词： 车联网，资源分配，分布式频谱接入，深度强化学习，多智能体强化学习。

Abstract

Vehicle-to-everything (V2X) communication will be an indispensable part of the future intelligent road traffic system. V2X technology links the participants in the road traffic system, and can thus improve traffic efficiency, reduce traffic accidents, enhance safety, and enrich driving experience. Since there are many types of vehicle user equipments (VUEs) in the V2X system, their quality-of-service (QoS) requirements can be different. In order to meet the diverse QoS requirements, it necessiates reasonable spectrum resource allocation. Nonetheless, the conventional centralized optimization based methodologies face significant challenges, since the high mobility of vehicles generates fast time-varying channels and the scarcity of spectrum resource limits the feasibility of central controller's global information acquisition. Therefore, in order to tackle the complex and dynamic environment, it is crucial to design decentralized spectrum access mechanism for the vehicular networks. At the same time, artificial intelligence (AI) technology, e.g., deep reinforcement learning (DRL), which opens up a novel data-driven design paradigm, has been increasingly exploited in the field of wireless communication in recent years. This thesis studies the design of decentralized spectrum access mechanism in cellular V2X (C-V2X) systems, where vehicle-to-infrastructure (V2I) VUEs coexist with vehicle-to-vehicle (V2V) VUEs. Aiming at maximizing the throughput of V2I-VUEs while meeting the latency and reliability requirements of V2V-VUEs, in this thesis we propose two decentralized spectrum access algorithms based on the multi-agent reinforcement learning (MARL).

Firstly, by formulating the optimization problem of vehicular network spectrum access as a decentralized partially observable Markov decision process (Dec-POMDP), we propose a fully decentralized spectrum access algorithm based on MARL, in which agents work independently and learn to collaborate. In the proposed algorithm, the agent optimizes the joint selection of sub-channel and transmission power level, using local environment observation that includes no channel state information (CSI). Through the delicate design of reward function and training mechanism, agents learn to form implicit collaborative behaviors. In order to tackle the complex

vehicular network environment and improve the learning performance, the proposed algorithm integrates a series of advanced DRL techniques. Additionally, in order to handle the non-stationarity induced by multi-agent concurrent learning, the algorithm incorporates hysteretic Q-learning and concurrent experience replay trajectories (CERT) to stabilize the training process. Besides, we incorporate the approximate regretted reward (ARR) to alleviate the unstable reward estimation problem caused by shifting environment dynamics in the vehicular networks. Simulation results validate the performance advantages of the proposed algorithm over baselines in both metrics of V2I-VUEs' sum throughput and V2V-VUEs' packet delivery ratio. Besides, the algorithm also exhibits good robustness and scalability.

In order to achieve better collaboration, in this thesis we also introduce the inter-agent communication mechanism into MARL. Considering that the agents not only learn effective policies for channel access, but also learn emergent communication protocols, we propose a decentralized spectrum access algorithm that achieves explicit collaboration by incorporating the inter-agent communication mechanism. In the proposed algorithm, each agent consists of an action selector module and a message generator module: the action selector decides the spectrum access action, and the message generator produces the interactive message. Both modules are implemented with a deep neural network (DNN) and share the parameters. In order to achieve end-to-end training of the message generator module, a discretize/regularize unit (DRU) is introduced, which regularizes the output of the message generator during the training phase and discretizes during the testing phase, so that the parameters of the message generator module can be updated by the gradient back-propagation of the loss function to achieve end-to-end training. To reduce training cost, all agents share the same network, and each agent can exhibit diverse behavior policies since they obtain different environment observations as input. Finally, the simulation results verify the effectiveness of the incorporated inter-agent communication mechanism, and the proposed algorithm can achieve better performance than the fully independent MARL based algorithm without communication, indicating that the communication mechanism can effectively improve the collaboration effect among the agents.

Keywords: vehicular network, resource allocation, decentralized spectrum access, deep reinforcement learning, multi-agent reinforcement learning.

缩写、符号清单、术语表

英文缩写	英文全称	中文全称
V2X	Vehicle-to-everything	车联网
QoS	Quality-of-service	服务质量
AI	Artificial Intelligence	人工智能
DRL	Deep Reinforcement Learning	深度强化学习
C-V2X	Cellular V2X	蜂窝车联网
VUE	Vehicle User Equipment	车辆用户设备
V2I	Vehicle-to-infrastructure	车对设施通信
V2V	Vehicle-to-vehicle	车对车通信
MARL	Multi-agent Reinforcement Learning	多智能体强化学习
Dec-POMDP	Decentralized Partially Observable Markov Decision Process	分布式部分可观测马尔可夫决策过程
CERT	Concurrent Experience Replay Trajectory	并发经验回放轨迹
ARR	Approximate Regretted Reward	近似遗憾奖励
DNN	Deep Neural Network	深度神经网络
DRU	Discretize/regularize Unit	离散/正则处理单元
3GPP	3rd Generation Partnership Project	第三代移动通信伙伴项目
DSRC	Dedicated Short Range Communication	专用短程通信
LTE	Long Term Evolution	4G长期演进
NR	New Radio	5G新空口
V2P	Vehicle-to-pedestrian	车对行人通信
V2N	Vehicle-to-network	车辆和网络通信
AoI	Age-of-information	信息年龄
DQN	Deep Q-network	深度Q网络

SARL	Single-agent Reinforcement Learning	单智能体强化学习
MDP	Markov Decision Process	马尔可夫决策过程
CommNet	Communication Neural Net	通信神经网络
RIAL	Reinforced Inter-Agent Learning	强化智能体间通信
DIAL	Differentiable Inter-Agent Learning	可微智能体间通信
ATOC	Attentional Communication	注意力通信
SchedNet	Schedule Network	调度网络
TarMAC	Targeted Multi-Agent Communication	目标多智能体通信
IS	Intention Sharing	意图共享
NLP	Natural Language Processing	自然语言处理
CIC	Causal Influence of Communication	通信的因果影响
CSI	Channel State Information	信道状态信息
S-SPS	Sensing based Semi-persistent Scheduling	基于感知的半持续调度
RSRP	Reference Signal Received Power	参考信号接收功率
RSSI	Reference Signal Strength Indicator	参考信号强度指示
RRC	Resource Reselection Counter	资源重选计数器
FMDP	Finite MDP	有限马尔可夫决策过程
GPI	Generalised Policy Iteration	广义策略迭代
TD	Temporal Difference	时间差分
DDPG	Deep Deterministic Policy Gradient	深度确定性策略梯度
PPO	Proximal Policy Optimization	近端策略优化
SG	Stochastic Game	随机博弈
MADDPG	Multi-agent DDPG	多智能体DDPG
MAPPO	Multi-agent PPO	多智能体PPO
BS	Base Station	基站
SINR	Signal to Interference plus Noise Ratio	信干噪比
MINLP	Mixed Integer Nonlinear Programming	混合整数非线性规划
RNN	Recurrent Neural Network	循环神经网络

GRU	Gated Recurrent Unit	门控循环单元
IL	Independent Learner	独立学习者
MAB	Muti-armed Bandit	多臂赌博机
MLP	Multi-layer Perceptron	多层感知机

插　图

表　格

目　　录

1 绪论

随着国民经济的迅速发展以及道路交通基础设施的完善，汽车作为重要的出行交通工具，在我国的保有量正迅速提高，根据国家统计局的数据显示，截至2020年，我国的私人汽车拥有量已达到24291万辆[1]。车联网通过先进的通信、感知、控制技术，将道路交通的各参与实体联结，其将有助于提高道路交通效率、增强安全减少事故，并且通过提供丰富的车载信息来提升驾驶体验[2]。车联网产业近年来处于高速发展状态，据相关产业调研机构数据[3]，2020年，我国车联网行业渗透率已达48.8%，超过全球车联网行业平均渗透率，用户规模约为13713万辆，且预计2025年中国车联网行业渗透率将超过75%，用户规模将超过3.8亿辆。随着国家对智慧交通政策的推广[4]，用户规模将持续扩大，车联网的市场需求将不断增长，车联网将逐渐普及。

1.1 研究背景

车联网是汽车、电子、信息通信、交通运输和交通管理等行业深度融合的新型产业形态[5]。车联网通信（Vehicle-to-everything, V2X）被认为是未来智慧交通体系的关键技术之一，其核心目标即是为车辆和道路基础设施，以及所有参与道路交通的实体单位，提供安全可靠的无线连接。通过及时可靠的信息交互，实现交通系统的整体协作，有助于提高道路安全、交通效率以及汽车上的娱乐体验。 当前主要有两种车联网候选技术方案，分别是3GPP（3rd Generation Partnership Project）推出的基于蜂窝网络的车联网通信（Cellular V2X, C-V2X）和基于IEEE 802.11p的专用短程通信（Dedicated Short Range Communication, DSRC）。由于C-V2X技术能够保证更好的覆盖范围和服务质量保障，能够提高频谱利用效率，当前更受学术界和工业界的瞩目[2,6]。C-V2X是基于3GPP统一标准的车联网通信技术，依托于3GPP长期积累的蜂窝移动通信标准优势，C-V2X标准制定已经迭代了数个版本，主要包括基于4G LTE（Long Term Evolution） 移动通信技术演进形成的 LTE-V2X/LTE-eV2X 技术，以及最新的基于 5G NR（New Radio）平滑演进形成的NR-V2X技术。随着越来越多先进技术特性的引入，C-V2X技术将作为关键使能技术，支撑诸如环境感知、自动驾驶、

流量调控等对通信极高要求的复杂任务，最终促进道路统一协同体系的形成。C-V2X主要定义了四种通信模式，分别为车对行人通信（Vehicle-to-pedestrian, V2P）、车辆和网络通信（Vehicle-to-network, V2N）、车辆和车辆通信（Vehicle-to-vehicle, V2V）以及车辆对设施通信（Vehicle-to-infrastructure, V2I）。随着科技发展，未来道路上车辆将不再只是单纯的交通工具，为提升驾驶体验，越来越多的娱乐、交通相关的车载应用将出现，而这些车载应用主要承载于V2I和V2V通信模式。如流媒体服务和道路地图更新这样的非交通安全相关应用，通常需要较高的数据传输速率和频繁的访问接入，可以基于V2I通信模式进行[7]；此外，对安全性要求较高的应用，如自动驾驶、队列驾驶等，需要在一定范围内的车辆之间以低延迟和高可靠性进行实时信息交换，可以通过V2V通信模式承载[8]。车联网通信中可能同时存在不同服务质量（Quality-of-service, QoS）需求的用户，如此前描述的高传输速率需求，以及低时延高可靠需求。同时满足不同QoS需求对于频谱资源管理和接入机制的设计提出了严苛的要求。文章[9]基于传统的优化算法思路，针对车联网环境中不同网络负载情况，以保障不同优先级车辆QoS为目标，提出了一种功率控制和传输模式选择的联合优化算法。在文章[9]中定义的中心式工作模式中，由中央基站为车辆统一分配频谱资源。然而，由于车联网中车辆的高移动性导致信道具有快速时变特性，限制了中央控制器获取全局信息的可行性。此外，由于车联网频谱资源比较稀缺，通过反馈信道获取信息会造成额外的信道负荷，因而进一步限制了传统基于优化的中心式资源分配方法在车联网中的应用[10]。因此，为了适应愈发复杂的应用环境，需要重新思考设计模式以提出更灵活、动态、可扩展的车联网分布式频谱接入及资源分配机制。

1.2 国内外研究现状

1.2.1 强化学习应用于车联网频谱资源分配

近年来，深度强化学习（Deep Reinforcement Learning, DRL）在许多涉及决策的领域，如游戏、控制和金融等，取得了令人瞩目的成绩[11]，这也激发了将DRL技术应用于解决无线通信领域中问题的热潮。使用强化学习进行决策，无需环境的准确模型。智能体在与环境的不断交互中，通过试错与探索来学习如何最大化奖励。对于无线网络系统优化的问题，如果基于传统的优化思路，则需要对系统进行准确的建模，例如需要获取信道状态信息等，而这通常会引入额外的信令开销，这使得传统优化方法受到了一定的限制。强化学习相对于传统优化方法灵活、动态的特点，使得其近年来逐渐受到学界的重视。当前强化学习已经在无线通信的许多方面取得了显著进展[12]，例如网络切片管

理[13]、信道编码设计[14]、资源分配[15]等。对于非车联网的通信环境，文章[16–19]已经展现了使用DRL技术来设计分布式频谱接入算法的潜力，而目前也有一些开创性的工作将DRL技术拓展应用于车联网中频谱接入机制的设计。例如，文章[20]以满足V2V链路在单播和多播通信模式下的时延约束为目标，提出了基于DRL的车联网分布式频谱资源分配机制。在文章[21]中，作者研究了C-V2X通信中传输模式选择和资源分配的联合优化问题，并以同时满足V2I和V2V用户的QoS要求为目标，提出了一种基于DRL的分布式接入算法。文章[22]研究了V2V网络中基于信息年龄（Age-of-information, AoI）感知的频谱资源管理问题，并基于DRL技术提出了一种分布式频谱资源分配及数据包调度算法。在上述文章[16–22]中，均采用了著名的深度Q网络（Deep Q-network, DQN）算法，其将Q-学习与深度神经网络（Deep Neural Network, DNN）相结合，使用DNN作为函数近似器来逼近值函数，将输入状态映射到动作的价值估计，以进行决策[23]。

然而，包括DQN在内的传统强化学习算法通常是专为静态环境所设计的，因此当环境动态特性发生变化时，智能体学习到的策略将不再适用，算法性能会急剧下降[24]。然而环境动态特性的改变在现实世界中是十分常见的，例如车联网中车辆的位置和信道特性等都是在不断发生改变的。环境动态特性的分布变化对强化学习中的奖励估计提出了直接的挑战，其可能会误导智能体学习策略产生偏差。例如，当智能体获得的奖励增加时，智能体很难区分这种增加的结果是缘于此前自己决策的动作产生了好的效果，还是仅仅由于环境内在特性发生变化，如信道质量变好，使得决策难度降低，智能体本身就更容易获得高的奖励，而这种获得奖励的提升与其行为策略是无关的。然而，现有工作[16–22]很少考虑无线网络中环境动态特性变化引起的奖励估计偏差问题，这会使得算法的稳健性受到挑战。

将强化学习应用于车联网频谱接入机制的设计中存在的另一个问题是，每个单独的智能体获得的奖励不仅取决于当前环境和它自身的行动策略，还取决于所有其它智能体的联合策略，即这是一个多智能体强化学习（Multi-agent Reinforcement Learning, MARL）问题[25]。如果直接将单智能体强化学习（Single-agent Reinforcement Learning, SARL）算法，如著名的DQN，扩展到MARL设置，如文章[16–21]，由多个智能体并发探索学习引起的非平稳特性会严重阻碍训练过程并降低算法性能[26]。文章[22]可以算作一个例外，在该文章中其将频谱接入问题建模为一个中心式的决策问题，即一个SARL问题，随后通过将该中心式决策问题分解为每个用户的马尔可夫决策过程（Markov Decision Process, MDP），最终得以实现分布式决策。

为了解决上述多智能体同时学习带来的非平稳性问题，现在已经有一些工作直接从MARL的角度研究车联网信道接入问题。例如，文章[27]研究了多智能体行车系统中车

辆的最优接入控制问题，并提出了一种结合统计学习方法和动态规划方法的分布式接入算法。然而，由于该算法使用了基于表格的动态规划方法，状态空间必须被离散化，这限制了其在具备高维状态空间问题中的适用性。在文章[28]中，作者提出了基于DQN的分布式算法来优化车联网的频谱资源和功率分配联合优化问题，并提出了基于指纹的重放缓冲机制来解决非平稳性问题。文章[29]同样提出基于指纹重放缓冲机制的DQN算法来解决C-V2X车队行驶系统的频谱资源分配问题。在文章[30]中，作者提出了一种基于MARL的算法来优化拥塞场景的V2V通信频谱资源分配，其中作者提出基于视图的位置分布来作为智能体的特殊状态表征来应对非平稳特性。尽管这些工作[28,29]均呈现出一定的性能提升，但仍存在一些缺陷，如文章[28]和[29]没有考虑环境动态特性改变对算法性能的影响，而文章[30]中提出的状态表征仅适用于于车辆在单一方向移动的场景。

1.2.2 引入通信机制的多智能体强化学习

为了更好地使智能体完成协作任务，一种比较直观的思想是使智能体之间能够彼此交互信息。通过信息交互，各智能体够对其余智能体更好地进行建模，根据其余智能体的策略做出更适宜的决策。在多智能体强化学习算法设计中引入通信机制是当下的一个研究热点，从最初人为指定通信协议（包括人工设计通信内容、通信时机、通信对象等），到通过深度学习技术端到端学习通信协议，近年来，结合通信协议学习的MARL领域已经涌现出了一批杰出的工作[31]。

文章[32]和文章[33]率先将通信机制引入MARL算法设计中。文章[32]提出了CommNet（Communication Neural Net）算法，每个智能体维护一个神经网络，并且均能够接入一个承载连续信息的广播信道，通过该广播信道获取其余智能体神经网络的中间层隐状态，以此来实现信息交互。文章[33]则分别提出了RIAL（Reinforced Inter-Agent Learning）算法和DIAL（Differentiable Inter-Agent Learning）算法：RIAL算法预定义了离散的通信消息集合，将选择通信消息作为动作，使用DQN进行训练；而DIAL算法在训练阶段发送连续消息，直接使用梯度对产生消息的网络参数进行端到端训练，在执行阶段才对消息进行离散化操作，充分发挥了中心式训练的优势。

文章[32,33]中智能体采用了预先定义的通信结构，即统一向周围广播消息，当智能体数目增多时，这种结构可能难以区分出何为有价值的信息。为了解决这一问题，文章[34]提出了ATOC（Attentional Communication）算法，通过引入注意力机制，来学习判断真正需要进行通信的时机，以及如何对接受到的信息进行整合。文章[35]考虑现实通信系统中通信带宽受限的场景，并且智能体彼此共享广播信道，因此需要对智能体间的通

信进行调控，该工作提出了SchedNet（Schedule Network）算法，通过学习评估智能体各自对环境观测结果的重要性，来决定允许哪些智能体接入信道并广播消息。文章[36]提出了TarMAC（Targeted Multi-Agent Communication）算法实现有目标的通信交互，其引入注意力机制来学习发送的消息内容，以及关注哪些发送方。并且该工作指出，在执行动作之前，通过多轮的消息交互能有效提高协作表现。文章[37]则提出了名为意图共享（Intention Sharing, IS）的通信机制来增强多智能体协作，智能体通过对环境动力和其它智能体策略进行建模，产生其自己未来一段时间内预计的行为轨迹，即所谓的“意图”。在该算法中智能体通过将“意图”数据压缩后进行通信交互，并通过注意力机制来表示各步骤的相对重要性。

在多智能体系统中通过深度学习技术训练得到通信协议的这一过程，可以类比为一种“机器语言”的诞生，因此也有研究人员从自然语言处理（Natural Language Processing，NLP）的角度来研究该问题，试图将人类语言和“机器语言”联系起来。为了对“机器语言”进行定性定量的分析，确立评价指标是非常必要的。文章[38]对常用的几种衡量指标进行了比较分析。该工作还证明了通过学习得到的通信协议并不一定会显式地影响其它智能体的行为，即智能体只学会了“说”，而没学会“听”，因此该论文提出采用通信的因果影响（Causal Influence of Communication, CIC）来衡量发送的消息对其它智能体的影响程度。对“机器语言”的理解分析，目前仍然是一个开放的问题，主要存在两方面挑战[38,39]：（1）可解释性分析，即理解“机器语言”的语义模式；（2）转化为人类可理解的语言模式。通过对“机器语言”进行剖析，有助于加深人们对自然语言的理解，这一领域还有很多可供挖掘的方向。

经典的通信系统以在噪声信道中尽可能准确地传输信息为目标进行设计，而在结合了通信机制的MARL框架中，通信的直接目的是为了实现更好地协作。结合了通信机制的MARL框架为多用户通信系统的设计开辟了一个新方向，通过深度强化学习技术端到端训练，智能体可以学会如何高效地进行通信，以更好地辅助智能体协作完成任务[40]。如今已有一些前沿的工作在这方面开展研究，文章[40,41]研究了噪声信道下，多智能体协作场景的联合学习通信框架，通过将发送的消息视作动作的一部分，并使用DRL技术进行优化，智能体可以学会如何有效地进行通信来更好地完成协作任务。该文章提出的框架同样适用于传统的分别考虑任务协作与通信机制设计的场景，例如单独使用DRL优化信源-信道联合编码设计。相对于传统方案设计，该框架依然具备一定的性能增益。文章[42]将MARL算法设计应用于车联网频谱接入控制问题，为了减小信令负荷，其使用DNN并结合量化技术来压缩信道状态信息（Channel State Information, CSI），中心基站

根据接收到的各车辆压缩CSI来进行决策，为各车辆分配频谱资源。更进一步，该作者提出了一种分布式决策机制，即中心基站使用DNN将接收到的CSI信息进行整合，广播给各车辆，车辆再基于MARL技术进行分布式决策。由于该框架中的CSI信息将会参与各智能体的决策过程，因此也可以看作引入了通信机制来辅助协作的MARL算法。截至目前，将结合通信机制的多智能体强化学习算法应用于无线通信系统的设计还是一个比较开放的领域，在诸多具体的场景中，如通用的联合编码设计、协作机制设计，乃至于协议可解释性分析等，仍有相当多的工作可以挖掘。

1.3 现有标准方案

3GPP在Release 16中为5G NR V2X定义了两种新的信道资源选择模式，即Mode 1和Mode 2，这两种模式可以看作4G LTE V2X中所定义的资源选择模式Mode 3 和Mode 4的扩展，区别在于LTE V2X只支持广播通信模式，而NR V2X同时支持单播、组播以及广播模式[43]。

与LTE V2X中的Mode 3类似，在NR V2X中的Mode 1模式下，基站为V2V通信进行频谱资源的分配和管理，因此车辆必须位于基站覆盖范围下，Mode 1属于中心式的分配策略。由于文章篇幅所限，此处侧重于介绍分布式的频谱资源分配方案，关于Mode 1的更多内容可参见文章[44]。

当车辆使用NR V2X Mode 2时，同LTE V2X Mode 4一样，用户可从资源池中自主选择频谱资源，Mode 2是分布式的工作模式，因此无需处于基站覆盖范围即可工作。Mode 2和Mode 4的区别在于调度策略，Mode 4使用基于感知的半持续调度策略（Sensing based Semi-persistent Scheduling, S-SPS），而Mode 2既可工作于半持续调度，也可使用动态调度策略。动态调度策略为每次传输都重新进行资源选择，仅可为重传预留资源，而半持续调度机制则为未来的若干次传输预留资源。

Mode 2模式中，动态调度和半持续调度几乎基于相同的流程进行资源选择，即基于感知的资源选择机制，其基于安全消息以固定的数据包大小周期性发送这一基本假设进行设计[45,46]。该基于感知的资源选择过程主要包括两个步骤：（1）从选择窗口中确定所有候选资源；（2）从候选资源中随机进行选择。如下图1.1所示，可根据选择窗口中的每一个资源块过去一段时间（即感知窗口）内的平均干扰功率大小来判断其是否适宜作为候选资源。

步骤（1）确定候选资源主要是通过排除选择窗口中不适宜的资源块实现的，具体实现如下：

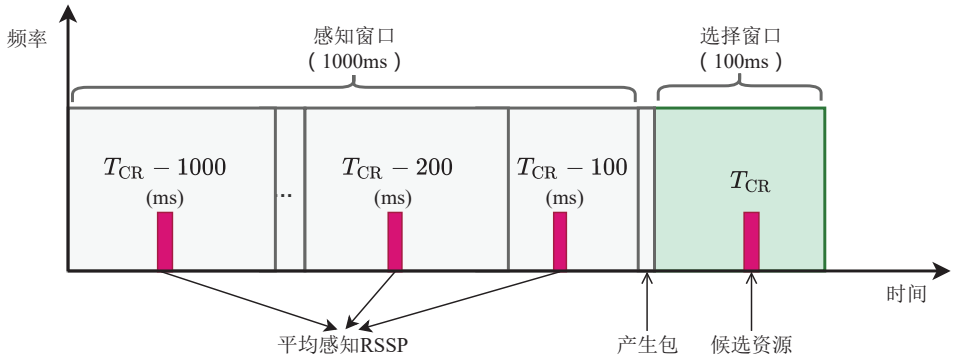


图 1.1 Mode 2基于感知的资源选择示意图（以发送周期为100ms为例）

1) 当一个用户需要发消息时（首次发送，或者重选），假设其发送消息频率为λHz，则最大允许延迟为$\frac{1000}{\lambda}$ms（例如当发送消息频率为10Hz时，最大允许延迟即为100ms），该最大延迟通常对应选择窗口大小，选择窗口内的所有子信道均为候选资源。

2) 用户会感知过去1000ms（即感知窗口大小）内的所有资源，以判断哪些资源在接下来的传输中是最合适的。该用户创建一个表格L_A，该表格包含所有可以保留的资源（即步骤i中选择窗口中的资源），除了以下例外：

   - 该用户在感知窗口内从别的用户接收并正确解码控制信息，表明别的用户会在选择窗口内使用该资源。
   - 该资源块过去的平均接收功率，由参考信号接收功率（Reference Signal Received Power, RSRP）表征，超过了门限值P_{th}。$P_{\mathrm{th}}=-128+2(a\cdot 8+b)$dBm，其中$a,b\in\{0,1,...,7\}$对应发射机和接收机的优先级。由于假设V2V发送的安全消息为周期性发送，因此若以发送周期为100ms为例，选择窗口中T_{CR}时刻的候选资源块在感知窗口内对应的时刻分别为$T_{\mathrm{CR}}-j\cdot 100, j\in\{1,2,\ldots,10\}$，对应平均RSRP感知结果如下：

$$\overline{\mathrm{RSRP}}=\frac{\sum_{j=1}^{10}\mathrm{RSRP}_{T_{CR}-100\cdot j}}{10} \tag{1.1}$$

     其中$\mathrm{RSRP}_{T_{CR}-100\cdot j}$为过去各时刻的RSRP感知结果。

   此外，如果该用户过去在该资源块上进行过传输，也会将其排除。当步骤ii执行完成时，表格L_A必须至少包含选择窗口中资源块总数的20%，否者将门限值P_{th}提高3dB。

3) 用户将表格L_A中经历了最低平均参考信号强度指示（Reference Singal Strength Indicator, RSSI）的资源块添加到列表L_C中，表格L_C中资源数应该等于选择窗口中资源总数

的20%。类似地，平均RSSI的计算过程如下：

$$\overline{\mathrm{RSSI}} = \frac{\sum_{j=1}^{10} \mathrm{RSSI}_{T_{\mathrm{CR}}-100\cdot j}}{10} \tag{1.2}$$

随后用户即可执行步骤（2），从列表L_C中随机选择一个资源块进行传输。资源选择完成后，考虑到用户将周期性发送消息，基于半持续调度思想，用户可预留该选择的资源以备未来的若干次传输，资源可预留次数取决于资源重选择计数器（Resource Reselection Counter, RRC）的大小。每完成一次传输时，重选计数器减一，当RRC减小为0时，其有一定概率P选择在此前资源上继续传输，以概率$1-P$重新进行资源选择，P值可设置为$[0, 0.8]$。另外RRC的初始大小设置取决于用户的消息传输周期，定义传输间隔为资源预留周期（Resource Reservation Period）P_{rsvp}，P_{rsvp}为1ms到1000ms范围内的一预设值，如果P_{rsvp} >100ms，则初始RRC值在区间$[5, 15]$内随机选择；如果P_{rsvp} <100ms，则RRC值在区间$\left\lceil 5\times\frac{100}{\max(20,P_{\mathrm{rsvp}})}, 15\times\frac{100}{\max(20,P_{\mathrm{rsvp}})}\right\rceil$内随机选择。

以上简要介绍了NR Mode 2资源选择的基本流程，更细致的介绍可参见文章[43–46]。

1.4 论文主要贡献与结构安排

本学位论文主要研究车联网中的分布式频谱接入算法设计，考虑城市道路中V2I用户和V2V用户共存场景下，以最大化V2I用户总吞吐量同时满足V2V用户时延可靠性需求为优化目标，基于多智能体强化学习分别提出了两个算法。本学位论文的主要贡献如下：

1) **完全独立的分布式频谱接入算法。**通过将车联网频谱接入优化问题建模为分布式部分可观测马尔可夫决策过程（Decentralized Partially Observable Markov Decision Process, Dec-POMDP），本文基于多智能体强化学习设计了一种完全分布式的频谱接入算法。在该算法设计中，V2V用户可在仅需局部环境观测且无需信道状态信息的条件下联合优化子信道和传输功率的选择，并通过奖励函数和训练机制的设计，学习形成隐式的协作模式。该算法结合了众多先进的DQN算法改进，并且为了应对多智能体并发学习所引起的非平稳性，该算法还引入了滞后学习机制和并发经验回放机制，来实现更好的训练效果。为了解决环境动态变化导致智能体难以训练的问题，该算法还引入了一种近似遗憾奖励机制来实现更准确的训练效果评估。最后，仿真实验验证了该算法的性能优势，且该算法具备一定的稳健性和扩展性。

2) **引入通信机制的分布式接入算法。**为了实现更好的协作，本文还将通信机制引入多智

能体强化学习中，提出了一种通过结合智能体间通信机制实现显式协作的分布式频谱接入算法。在该算法中，智能体由动作选择模块和消息产生模块组成：动作选择模块完成频谱接入动作的选择，而消息产生模块则负责生成交互信息。动作选择模块和消息产生模块均由神经网络实现，且共享网络参数。为了实现消息产生模块的端到端训练，该算法引入了离散/正则处理单元，其在训练阶段对消息产生模块的输出正则化，而在测试阶段离散化，可以使得消息产生模块的参数可以通过损失函数进行梯度回传更新，实现端到端训练，且减小在测试阶段的性能损失。在训练阶段，所有智能体共享网络参数，以减小训练开销。最终，仿真结果验证了引入通信机制的有效性，能够有效提升智能体间的协作效果。

本学位论文正文总共包含五个章节，具体内容安排如下。

第一章为绪论，首先介绍了车联网通信的研究背景，然后针对车联网频谱接入方案的研究现状，分别介绍了使用强化学习设计车联网频谱资源分配方案的相关工作，以及将通信机制引入多智能体强化学习这一前沿领域的最新进展。此外，本章还介绍了3GPP现有标准协议中的车联网频谱资源分配方案。通过对现有工作进行详尽的综述调研，引出了本论文的研究内容。

第二章主要介绍强化学习，及多智能体强化学习相关的背景知识，为后文基于多智能体强化学习进行问题建模和算法设计确立基本框架，将在该章节对后文算法设计过程中涉及到的必要概念进行阐述。

本论文主要研究车联网城市道路场景中V2I用户和V2V用户共存场景下的分布式频谱接入算法设计，分别在第三章和第四章基于多智能体强化学习提出两种算法。

第三章首先根据V2I用户和V2V用户不同业务需求对所研究的问题进行数学形式上的优化问题建模，确定了整体优化目标：最大化V2I用户的总吞吐量，同时满足V2V用户的时延可靠性要求。随后基于多智能体强化学习框架对优化问题进行了相应的转化建模，并提出了一种完全分布式的频谱接入算法。通过在算法设计中引入多种先进的学习技术，该算法相对于对比基准实现了性能提升。

第四章将通信机制引入多智能体强化学习框架之中，通过在决策过程中进行信息交互，可以使智能体间实现更好的协作。在本章节所提出的算法中，智能体不仅需要决策频谱接入相关动作，同时还需要决定当前时刻需要发送的交互信息。智能体的动作选择模块和消息产生模块均使用神经网络实现，且共享参数，通过深度强化学习技术实现端到端联合优化。仿真结果表明，引入通信机制能够有效提高算法性能。

最后，第五章将对文章全文进行总结，并对未来可行的潜在拓展研究方向进行讨论。

2 强化学习基础

强化学习（Reinforcement Learning, RL），是人工智能领域中一类特定的机器学习问题。RL从统计学、控制理论和心理学等多学科交叉发展而来，是一个基于数学框架、由经验驱动的自主学习方法[47]。深度学习（Deep Learning, DL）作为机器学习研究中的重要领域，近年来随着硬件平台计算能力的长足发展，在图像、文本、语音等诸多应用领域取得了瞩目的成绩[48]。自然而然，人们同样会期望RL能够借助DL来解决以往难以处理的问题，例如直接读取像素来玩视频游戏等。谷歌Deepmind于2015年发表于Nature的文章[23]使用深度强化学习首次实现了在Atari游戏中达到与人类同等甚至更高的水平，向世人展现了DRL的巨大潜力。

本章节将首先简要介绍强化学习中的基本概念及算法，并在第二部分侧重于多智能体领域，介绍多智能体强化学习的基本建模框架及算法。

2.1 强化学习基础

2.1.1 马尔可夫决策过程

在一个强化学习系统中，智能体可以观察环境，并根据相应的环境状态做出动作。在智能体行动之后，环境反馈其奖励。强化学习通过与环境的交互来学习如何最大化奖励。下图 2.1展示了强化学习系统中智能体与环境交互的基本框架。

通常单智能体强化学习问题可在形式上描述为马尔可夫决策过程，即MDP。MDP可由元组$\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma \rangle$表示[31,49,50]，其中

- $\mathcal{S}$表示环境状态空间；
- $\mathcal{A}$表示智能体的动作空间；
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$：表示奖励函数；
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$：表示环境的状态转移概率；

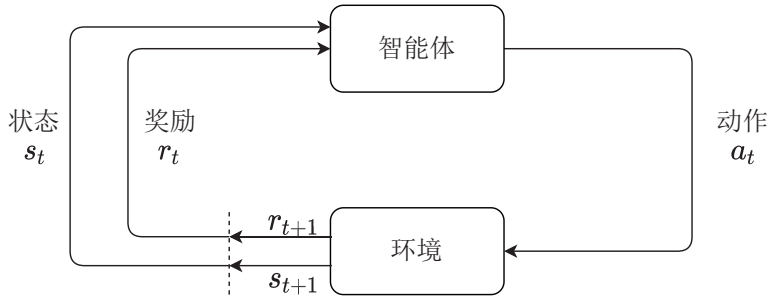


图 2.1 智能体-环境交互过程

- $\gamma \in [0,1]$表示奖励折扣系数。

如果状态空间$\mathcal{S}$和动作空间$\mathcal{A}$均为有限空间集合，则该MDP称为有限MDP（Finite MDP, FMDP）。MDP假设环境状态转移概率满足马尔科夫性，即下一时刻环境观察状态s_{t+1}和获得的奖励r_{t+1}，仅仅依赖于当前时刻的状态s_t和动作a_t，而不依赖于更早时刻的状态和动作。马尔可夫性是MDP模型对状态的额外约束，它要求当前环境状态必须包含可能对未来产生影响的所有信息。图 2.1中的交互过程表示在时刻t，智能体从环境观察到状态$s_t \in \mathcal{S}$，根据其行动策略选择了动作$a_t \in \mathcal{A}$，环境根据智能体动作相应反馈奖励$r_{t+1} = \mathcal{R}(s_t, a_t, s_{t+1}) \in \mathbb{R}$，并且环境状态将根据转移概率$P(s_{t+1}|s_t, a_t) = \mathcal{T}(s_t, a_t, s_{t+1})$转移至$s_{t+1} \in \mathcal{S}$。

MDP的环境由动力（Dynamics）刻画，对于FMDP，可以定义函数$p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0,1]$为MDP的动力：

$$p(s', r|s, a) = \Pr\left[s_{t+1} = s', r_{t+1} = r|s_t = s, a_t = a\right] \tag{2.1}$$

给定式（2.1），就可以相应推导出其它关于环境的相关内容，如给定“状态-动作”的期望奖励：

$$r(s, a) = \mathbb{E}\left[r_{t+1}|s_t = s, a_t = a\right] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s, a) \tag{2.2}$$

及状态转移概率：

$$p(s'|s, a) = \Pr\left[s_{t+1} = s'|s_t = s, a_t = a\right] = \sum_{r \in \mathcal{R}} p(s', r|s, a) \tag{2.3}$$

智能体根据对环境的观察结果来进行决策。在MDP中，策略可定义为从状态到动作的概率分布的映射，其策略$\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$可定义为：

$$\pi(a|s) = \Pr\left[a_t = a|s_t = s\right], s \in \mathcal{S}, a \in \mathcal{A} \tag{2.4}$$

如果动作集合为连续集合，则可以用概率分布来定义策略。而如果策略π 对于任意的$s \in \mathcal{S}$，均存在一个$a \in \mathcal{A}$，使

$$\pi(a'|s) = 0, \forall a' \neq a, a' \in \mathcal{A} \tag{2.5}$$

则该策略π为确定性策略。

2.1.2 强化学习基本概念

强化学习的核心概念之一为奖励，强化学习的目标通常为最大化长期累积奖励的期望。如果MDP刻画的是回合制任务，即某一回合在第T步将到达终止状态，这一整个回合的状态、动作和奖励序列构成了该智能体策略的轨迹。可定义时刻t（$t < T$）之后的回报（Return）G_t为未来奖励的和：

$$R_t = r_{t+1} + r_{t+2} + \ldots + r_T = \sum_{\tau=0}^{T-1} r_{t+\tau+1} \tag{2.6}$$

对于非回合制的连续性任务，由于其没有终止时间，因此为了避免式（2.6）中所定义回报形式得到无限累积奖励，可引入折扣的概念，定义折扣回报为：

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{\tau=0}^{+\infty} \gamma^\tau r_{t+\tau+1} \tag{2.7}$$

其中$\gamma \in [0, 1]$，其值大小反应了对未来奖励的重视程度，若$\gamma = 1$则等价于回合制无折扣任务回报。

不失一般性，强化学习的目标可表述为找到一个最优策略$\pi^\star$，其可以最大化所有状态的预期回报：

$$\pi^\star = \text{argmax}_\pi \, \mathbb{E}[R_t|\pi] \tag{2.8}$$

基于回报函数的定义，可进一步定义价值函数。几乎所有的强化学习算法都包含价值函数估计，即在给定状态（或状态-动作对）下对智能体表现好坏程度的估计。此处好坏程度可用预期回报来衡量，显然智能体未来可能获得的奖励取决于其行动策略，因此值函数的定义依托于特定策略。下面给出具体定义。

状态价值函数：状态价值函数$V_\pi(s)$表示从状态s开始，智能体按照策略π行动所获得的预期回报：

$$V_\pi(s) = \mathbb{E}_\pi \left[R_t | s_t = s \right] = \mathbb{E}_\pi \left[\sum_{\tau=0}^{+\infty} \gamma^\tau r_{t+\tau+1} | s_t = s \right] \tag{2.9}$$

状态-动作价值函数：状态-动作价值函数$Q_\pi(s,a)$表示在状态s采取动作a后，智能体按照策略π行动所获得的预期回报：

$$Q_\pi(s,a)=\mathbb{E}_\pi\left[R_t|s_t=s,a_t=a\right]=\mathbb{E}_\pi\left[\sum_{\tau=0}^{+\infty}\gamma^\tau r_{t+\tau+1}|s_t=s,a_t=a\right] \tag{2.10}$$

状态价值函数和状态-动作价值函数可使用Bellman期望方程来刻画相互转换关系，如下所示[51]:

- 用状态-动作价值函数表示状态价值函数：

$$V_\pi(s)=\sum_a\pi(a|s)Q_\pi(s,a),s\in\mathcal{S} \tag{2.11}$$

- 用状态价值函数表示状态-动作价值函数：

$$\begin{aligned}Q_\pi(s,a)&=r(s,a)+\gamma\sum_{s'}p(s'|s,a)V_\pi(s')\\&=\sum_{s',r}p(s',r|s,a)\left[r+\gamma V_\pi(s')\right],\quad s\in\mathcal{S},a\in\mathcal{A}\end{aligned} \tag{2.12}$$

基于式（2.11）和式（2.12），使用代入法消除其中一种价值函数，可得到状态价值函数和状态-动作价值函数的递归表示：

- 用状态价值函数表示状态价值函数：

$$V_\pi(s)=\sum_a\pi(a|s)\left[r(s,a)+\gamma\sum_{s'}p(s'|s,a)V_\pi(s')\right],s\in\mathcal{S} \tag{2.13}$$

- 用状态-动作价值函数表示状态-动作价值函数：

$$Q_\pi(s,a)=\sum_{s',r}p(s',r|s,a)\left[r+\gamma\sum_{a'}\pi(a'|s')Q_\pi(s',a')\right],\quad s\in\mathcal{S},a\in\mathcal{A} \tag{2.14}$$

在环境动力（$p(s',r|s,a)$）已知的情况下，即可根据上述式（2.13）和式（2.14）使用动态规划的方式来逼近给定策略π下的期望值函数。

由于值函数反映了策略π的预期期望，因此可基于值函数的大小来定义最优策略。首

先定义最优状态价值函数为：

$$V^{\star}(s)=\max_{\pi} V_{\pi}(s), \quad s \in \mathcal{S} \tag{2.15}$$

最优状态-动作价值函数如下：

$$Q^{\star}(s,a)=\max_{\pi} Q_{\pi}(s,a), \quad s \in \mathcal{S}, a \in \mathcal{A} \tag{2.16}$$

对于一个特定MDP，可能存在多个最优策略。当然根据如上定义，这些最优策略都有相同的价值函数，因此任取一个最优策略考察不失一般性。其中一种最优策略为按照如下确定性策略方式选择动作：

$$\pi^{\star}(s)=\mathrm{argmax}_{a\in\mathcal{A}} Q^{\star}(s,a), \quad s \in \mathcal{S} \tag{2.17}$$

2.1.3 深度强化学习算法

为了从任意Q_{π}中找到$Q^{\star}$，可以采用广义策略迭代（Generalised Policy Iteration, GPI），其包括策略评估和策略改进两部分[51]。策略评估可通过Bellman期望方程的方式来迭代更新值函数估计。随着估计的改进，通过值函数贪婪地选择动作，自然可以得到改进的策略。策略评估和策略改进通常可交替进行，以加快收敛过程。

经典的策略评估方法包括蒙特卡洛方法和时间差分（Temporal Difference, TD）方法。TD学习是蒙特卡洛思想和动态规划思想的结合，与蒙特卡洛类似，TD学习直接从策略π经历的轨迹来更新其价值估计，而无需像动态规划一样需要准确的环境动态模型；而TD学习又吸纳了动态规划的自举思想，基于其自身学习的估计来递归更新，无需像蒙特卡洛方法一样等待回合结束。蒙特卡洛方法和TD学习应用于值函数更新的基本方式分别如下：

$$V(s_t) \leftarrow V(s_t)+\alpha\left[R_t-V(s_t)\right] \tag{2.18}$$

$$V(s_t) \leftarrow V(s_t)+\alpha\left[r_{t+1}+\gamma V(s_{t+1})-V(s_t)\right] \tag{2.19}$$

其中α为更新步长。蒙特卡洛的更新目标为R_t，即跟随时间t的实际奖励，而TD更新的目标是$r_t+\gamma V(s_{t+1})$，因为TD基于现有估计，所以称TD是一种自举（Bootstrapping）的学习方法。

使用时序差分更新方法来求解最优策略，按照学习的目标策略和实际产生状态转移数据的行动策略是否一致进行划分，基本算法有同策学习（On-policy）的SARSA（State-

Action-Reward-State-Action）算法和异策学习（Off-policy）的Q-learning算法。SARSA算法和Q-learning算法更新方式可统一为：

$$Q_\pi(s_t,a_t) \leftarrow Q_\pi(s_t,a_t) + \alpha\left[Y - Q_\pi(s_t,a_t)\right] \tag{2.20}$$

其中Y可理解为标准回归问题中的拟合目标。

SARSA算法得名于其更新涉及的样本元组$(s_t,a_t,r_{t+1},s_{t+1},a_{t+1})$，其单步时序差分更新目标为：

$$Y = r_t + \gamma Q_\pi(s_{t+1},a_{t+1}) \tag{2.21}$$

异策学习的Q-learning算法相对于同策算法更为流行，已成为最经典的基础算法之一。Q-learning从策略改进后的行为出发，其更新目标为：

$$Y = r_t + \gamma \max_a Q(s_{t+1},a) \tag{2.22}$$

经典的Q-learning算法使用表格来记录状态和动作相对应的Q值，然而在现实世界中的诸多任务中，状态的维度可能是巨大的，甚至存在状态为连续值的情况，此时再用表格来存储Q值显然不够现实。并且按照Q-learning的更新方式，所有状态都至少需经历过一次，才能计算出Q值，因此在庞大的状态空间中进行搜索将耗费很多时间。Q-learning面临的上述问题被人们称为维度灾难（Curse of Dimensionality）。为了解决维度灾难，研究者提出使用神经网络（Neural Network, NN）作为值估计器，即直接将状态作为神经网络的输入，计算出所有动作对应的价值，即可替代传统的状态价值表。使用NN作为函数估计器的另一优势在于，其能处理连续输入状态，并且对于相近的状态能产生泛化性，避免对整个状态空间都进行遍历。

DQN算法[23]便是将深度神经网络和Q-learning相结合，开创了深度强化学习领域的先河。为了训练神经网络，DQN借鉴了深度学习领域的成功经验，引入了经验回放缓存（Experience Replay Buffer）机制，其将智能体在与环境交互过程中产生的转移样本$e_t = (s_t,a_t,r_{t+1},s_{t+1})$存储进数据集合$D = \{e_1,e_2,\ldots,e_t\}$中，DQN在每次更新时，随机（均匀采样）从$D$中抽取一些之前的经验样本进行学习，通过打乱样本之间的相关性，提高了神经网络训练的稳定性。DQN可通过最小化在第i次迭代时的如下损失函数来进行更新：

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim \mathrm{U}(D)}\left[\left(r + \gamma \max_{a'} Q\left(s',a';\theta_i^-\right) - Q\left(s,a;\theta_i\right)\right)^2\right] \tag{2.23}$$

其中θ_i^-表示静态目标网络，其以较低的频率从θ_i复制参数，以使该损失函数的更新目标不

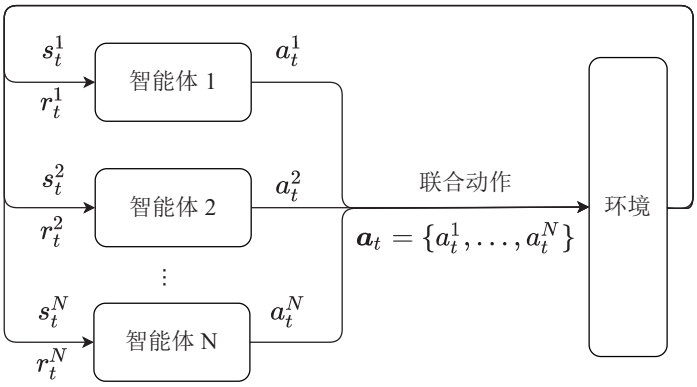


图 2.2 多智能体-环境交互过程

会一直变化，帮助稳定网络的训练过程。

除了基于值函数的DQN算法外，基于策略的算法，如深度确定性策略梯度（Deep Deterministic Policy Gradient, DDPG）算法[52]和近端策略优化（Proximal Policy Optimization, PPO）算法[53]等，近年来同样大放异彩，在诸多连续控制任务中取得了优异的表现。基于策略的强化学习算法直接维护参数化的策略π，因此可以直接输出动作，其也可以为动作分配概率分布，按照概率分布来执行动作。基于策略的算法相对于基于值函数的算法存在一个比较直接的优势，即基于策略的算法可以应用于动作空间为连续值的连续控制问题。由于篇幅限制，此处略去基于策略的算法更多细节。

2.2 多智能体强化学习基础

强化学习算法在单智能体场景取得了诸多成功的应用案例。在单智能体场景中，无须建模和预测环境中的其它智能体，但在现实世界中，同样存在很多涉及多个智能体间交互的重要应用场景。在多智能体场景中，存在智能体共同交互演化的过程，且智能体间可能存在协作、竞争和混合协作-竞争等不同关系，多智能体问题将比单智能体场景更加复杂。

图 2.2展示了基本的多智能强化学习系统，多智能体决策的联合动作使整个系统发生状态转移，环境将奖励返回给各智能体。本节接下来将介绍多智能体强化学习系统的建模框架及基本算法。

2.2.1 随机博弈

多智能体强化学习问题常用随机博弈（Stochastic Game, SG）进行描述，随机博弈又

称为马尔可夫博弈（Markov Game），其可通过元组$(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^n\}_{n\in\mathcal{N}}, \mathcal{T}, \{\mathcal{R}^n\}_{n\in\mathcal{N}}, \gamma,)$形式化描述，其中

- $\mathcal{N} = \{1, \ldots, n, \ldots, N\}$，表示智能体集合，其中$N$表示智能体个数；
- $\mathcal{S}$表示所有智能体共享的环境状态空间；
- $\{\mathcal{A}^n\}_{n\in\mathcal{N}}$表示智能体的动作空间集合，其中$\mathcal{A}^n$表示智能体$n$的动作空间。可定义联合动作空间为$\boldsymbol{\mathcal{A}} = \mathcal{A}^1 \times \mathcal{A}^2 \times \ldots \times \mathcal{A}^N$；
- $\mathcal{T} : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \Delta(\mathcal{S})$：给定所有智能体的联合动作$\boldsymbol{a} \in \boldsymbol{\mathcal{A}}$，环境从状态$s \in \mathcal{S}$转移到$s' \in \mathcal{S}$的概率；
- $\mathcal{R}^n : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to \mathbb{R}$：智能体$n$的奖励函数，给定联合动作$\boldsymbol{a}$，当环境从$s$转移至$s'$时，返回一标量实值奖励；
- $\gamma \in [0, 1]$：折扣系数。

随机博弈同样可展开序列化描述。在时刻t，环境状态为s_t，所有智能体同时采取动作a_t^n，组成联合动作$\boldsymbol{a}_t = \{a_t^1, a_t^2, \ldots, a_t^N\}$，环境接收到联合动作后根据转移概率$P(s_{t+1}|s_t, \boldsymbol{a}_t) = \mathcal{T}(s_t, \boldsymbol{a}_t, s_{t+1})$发生状态转移，并为每个智能体反馈即时奖励$r_{t+1}^n = \mathcal{R}^n(s_t, \boldsymbol{a}_t, s_{t+1})$。在随机博弈中，智能体的目标是寻找到一个策略$\pi^n \in \Pi^n : \mathcal{S} \to \Delta(\mathcal{A}^n)$，其能使如下期望累积折扣奖励最大化：

$$V^{\pi^n, \pi^{-n}}(s) = \mathbb{E}_{s_{t+1}\sim P(\cdot|s_t, a_t), a^{-n}\sim\pi^{-n}(\cdot|s_t)} \left[\sum_{t\geq 0} \gamma^t \mathcal{R}^n \left(s_t, \boldsymbol{a}_t, s_{t+1}\right) \mid a_t^n \sim \pi^n \left(\cdot \mid s_t\right), s_t = s \right] \tag{2.24}$$

此处使用上标$\cdot^n$和$\cdot^{-n}$来区分智能体n和集合$\mathcal{N}$中剩余的$N-1$个元素。式（2.24）表明在多智能体场景，每个智能体的价值函数不仅取决于其自身策略，还受到其它智能体策略的影响，这使得MARL算法和SARL算法存在本质上的不同。

随机博弈假设环境对所有智能体具有完全的可见性，但在实际生活的很多任务中，智能体并不具备全局观测的能力，其只能根据各自对环境的局部观测进行决策，此时相应可使用部分可观测马尔可夫决策过程Dec-POMDP进行建模。 Dec-POMDP是随机博弈的一种特例，其假设智能体无法直接获得确切的环境状态，而只能通过观测函数获得环境的部分观测结果。此外，Dec-POMDP假设所有智能体的奖励函数相同，即$\mathcal{R} = \mathcal{R}^1 = \ldots = \mathcal{R}^N$，因此常用来刻画多智能体强化学习中的协作场景。Dec-POMDP可

使用元组$(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^n\}_{n\in\mathcal{N}}, \mathcal{T}, \mathcal{R}, \gamma, \{\Omega^n\}_{n\in\mathcal{N}}, \mathcal{O})$形式化描述，其与随机博弈元组定义相比，额外增加了以下两项：

- Ω^n：智能体n的观测空间集合，可进一步定义联合观测空间为$\boldsymbol{\Omega} = \Omega^1 \times \Omega^2 \times \ldots \times \Omega^N$；
- $\mathcal{O} : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \Delta(\Omega)$：观测函数$\mathcal{O}(\boldsymbol{o}|\boldsymbol{a}, s')$ 表示给定联合动作$\boldsymbol{a} \in \boldsymbol{\mathcal{A}}$及环境转移到的下一状态$s' \in \mathcal{S}$，智能体接收到观测$\boldsymbol{o} \in \boldsymbol{\Omega}$的概率。

由于在Dec-POMDP中，智能体无法直接获得环境状态，而只能通过观测结果进行决策，因此智能体的策略表示变更为：$\pi^n \in \Pi^n : \Omega^n \to \Delta(\mathcal{A}^n)$。

2.2.2 多智能体强化学习算法

从博弈论角度出发[50]，若考虑不同智能体间的相互影响，式（2.20）中描述的单智能体Q函数更新表达式通过一定修改仍可成立。在第t次迭代中，对于智能体$n \in \mathcal{N}$，给定从经验回放缓存中采样得到的转移数据$\{(s_t, \boldsymbol{a}_t, r_t^n, s_{t+1})\}$，其Q值函数更新如下：

$$Q^n(s_t, \boldsymbol{a}_t) \leftarrow Q^n(s_t, \boldsymbol{a}_t) + \alpha \cdot \left[r_t^n + \gamma \cdot \textbf{eval}^n\left(\{Q^n(s_{t+1}, \cdot)\}_{n\in\mathcal{N}}\right) - Q^n(s_t, \boldsymbol{a}_t)\right] \tag{2.25}$$

其与式（2.22）的区别在于此处将$\max(\cdot)$操作更换为了$\textbf{eval}^n\left(\{Q^n(s_{t+1}, \cdot)\}_{n\in\mathcal{N}}\right)$，表明在MARL中，每个智能体不能只考虑自己，而必须通过考虑所有智能体的利益来评估（**eval**uate）时刻$t+1$阶段博弈的情况。随后，基于Q函数，可以求解（**solve**）得到智能体的最佳策略，即$\textbf{solve}^n\left(\{Q^n(s_{t+1}, \cdot)\}_{n\in\mathcal{N}}\right) = \pi^{n,*}$。类似的，根据Bellman最优准则，可得到$\textbf{eval}(\cdot)$算子和$\textbf{solve}(\cdot)$算子的如下关系：

$$\textbf{eval}^n\left(\{Q^n(s_{t+1}, \cdot)\}_{n\in\mathcal{N}}\right) = V^n\left(s_{t+1}, \left\{\textbf{solve}^n\left(\{Q^n(s_{t+1}, \cdot)\}_{n\in\mathcal{N}}\right)\right\}_{n\in\mathcal{N}}\right) \tag{2.26}$$

总之，算子$\textbf{solve}^n(\cdot)$返回在博弈均衡点处智能体n对应部分的最优策略（并不意味着其一定能获得最多奖励）；假设所有智能体均同意服从该均衡点的策略集合，则$\textbf{eval}^n(\cdot)$返回智能体n在该均衡点处所能获得的长期期望折扣奖励。

尽管如前所述，在MARL中，需要考虑不同智能体间策略的相互影响来设计算法，才可能得到最优解，但是也有一些研究工作直接将SARL算法扩展到MARL场景，并且取得了不错的结果，例如文章[54]将DQN算法与独立Q学习（Independent Q Learning, IQL）结合起来，每个智能体拥有独立的Q网络，独自采集数据并进行训练，通过调整奖励函数，可以完成完全协作、完全竞争和混合合作-竞争等不同设定下的任务。

MARL领域近来同样发展了很多基于策略的算法，如多智能体DDPG（Multi-agent DDPG, MADDPG）算法[55]和多智能体PPO（Multi-agent PPO, MAPPO）算法[56]等，将经典的单智能体策略算法结合中心式训练-分布式执行架构进行修改，在MARL场景取得了优异的成绩。 此外，多智能体强化学习算法还有从智能体间通信机制、信用分配以及训练机制等不同方向进行研究的工作，更多内容可参见文章[31,49,50]。

3 基于多智能体强化学习的分布式频谱接入算法

本章节研究C-V2X系统中V2I用户与V2V用户共存的接入控制问题。首先对子信道选择及发射功率控制的联合优化问题进行建模，随后基于分布式部分可观测马尔可夫决策过程框架从多智能体协作的角度研究该问题，并基于MARL提出了一种分布式频谱接入算法。通过在观测空间设计中去除了信道状态信息，本章节所提出的方法有助于减少C-V2X系统频谱资源分配中的信令开销。此外，为同时满足V2I用户和V2V用户的QoS要求，对奖励函数进行了针对性设计。由于车联网环境动态变化，智能体获得的奖励值分布会随着车辆的移动而改变，本章节提出了一个简单而有效的方法来解决该问题。仿真结果表明，本章节所提出的基于MARL的分布式频谱接入算法性能优于对比方案。本章节中部分重要数学符号及相关定义汇总见表3.1，以便查阅。

3.1 系统建模

3.1.1 场景模型

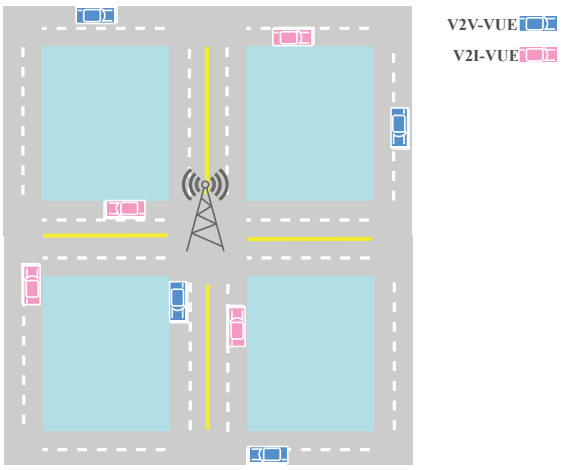


图 3.1 城市道路车联网场景示意图

考虑如图 3.1所示城市道路场景中的C-V2X通信系统，其中包括基站（Base Station,

表 3.1 部分重要符号汇总

符号	定义
$\mathcal{M}=\{1,2,\ldots,M\}$	V2I-VUE集合，M表示V2I-VUE总数
$\mathcal{N}=\{1,2,\ldots,N\}$	V2V-VUE集合，N表示V2V-VUE总数
$\mathcal{S}=\{1,2,\ldots,S\}$	子信道集合，S表示子信道总数
$\alpha_{m,s}$	V2I-VUE $m\in\mathcal{M}$是否占用子信道$s\in\mathcal{S}$
$\beta_{n,s}$	V2V-VUE $n\in\mathcal{N}$是否占用子信道$s\in\mathcal{S}$
$h^{\mathrm{I}}_{m,s}$	V2I-VUE m在子信道s上到关联基站间的信道增益
$h^{\mathrm{V}}_{n,l,s}$	V2V-VUE发射机m与对应接收机l间在子信道s的信道增益
$h^{\mathrm{V}}_{n,s}$	V2V-VUE n对基站在子信道s上的干扰信道增益
$h^{\mathrm{I}}_{m,l,s}$	V2I-VUE m对V2V-VUE接收机l在子信道s上的干扰信道增益
P^{I}_{m}	V2I-VUE m的发射功率
P^{V}_{n}	V2V-VUE n的发射功率
σ^2	噪声功率
$\gamma^{\mathrm{I}}_{m,s}$	V2I-VUE m在子信道s的上行信干噪比
$\gamma^{\mathrm{V}}_{n,l,s}$	V2V-VUE n在子信道 s上到接收机l的信干噪比
$R^{\mathrm{I}}_{m}(t)$	V2I-VUE m在时隙 t的上行链路吞吐量
$R^{\mathrm{V}}_{n,l}(t)$	V2V-VUE n在时隙 t到相应接收机l链路吞吐量
$R^{\mathrm{I}}_{\min}$	V2I-VUE的最低传输速率
$T_{\max}$	V2V-VUE消息发送周期
$L_n(t)$	V2V-VUE n在时刻t剩余待传输数据量
$\mathcal{P}=\{P^{\mathrm{V}}_{n}(t)\|n\in\mathcal{N}\}$	所有V2V-VUE的传输功率选择集合
$\mathcal{B}=\{\beta_{n,s}\|n\in\mathcal{N},s\in\mathcal{S}\}$	所有V2V-VUE的子信道选择集合
$o_n(t)$	智能体n在时间t接收到的环境观测
$a_n(t)$	智能体n在时间t选择的子信道和传输功率联合动作
$r(t)$	所有智能体在时间t获取的全局奖励
$\lambda_i, i\in\{1,2,3,4\}$	奖励权重
γ	折扣系数
ϵ	探索率
α	学习率
β	滞后率
θ	神经网络参数
δ	时间差分误差

BS）和以V2I或V2V模式运行的车辆用户设备（Vehicle User Equipment, VUE）。考虑V2I-VUE执行具有高吞吐量要求的上传任务，而V2V-VUE则周期性分发安全相关信息。

V2I-VUE和V2V-VUE的集合分别表示为$\mathcal{M} = \{1, 2, \ldots, M\}$和$\mathcal{N} = \{1, 2, \ldots, N\}$，其中$M$和$N$分别表示V2I-VUE和V2V-VUE的数量。此外，令子信道的集合表示为$\mathcal{S} = \{1, 2, \ldots, S\}$，其中$S$表示子信道数量。为提高频谱利用率，假设V2I-VUE与V2V-VUE共享频谱资源。分别使用0-1指示符$\alpha_{m,s}$和$\beta_{n,s}$表示V2I-VUE $m \in \mathcal{M}$ 和V2V-VUE $n \in \mathcal{N}$是否占用子信道$s \in \mathcal{S}$。此外，考虑每个VUE只能占用一个子信道，因此有$\sum_{s\in\mathcal{S}} \alpha_{m,s} \leq 1$以及$\sum_{s\in\mathcal{S}} \beta_{n,s} \leq 1$。

V2I-VUE m在子信道s上到关联基站间的信道增益表示为$h^{\mathrm{I}}_{m,s}$，V2V-VUE发射机n与对应接收机l之间在子信道s的信道增益表示为$h^{\mathrm{V}}_{n,l,s}$。此外，分别令$h^{\mathrm{V}}_{n,s}$和$h^{\mathrm{I}}_{m,l,s}$表示V2V-VUE n对基站，以及V2I-VUE m对V2V-VUE接收机l在子信道s上的干扰信道增益。在本章节建模中，信道增益包括大尺度衰落（包括路径损失和阴影）和小尺度衰落（瑞利衰落）。由于V2I-VUE与基站进行通信，因此为简化模型，并与文章[28]保持一致，本文假设V2I-VUE已经由基站预先分配了正交的子信道，并保持固定传输功率。

可得V2I-VUE m在子信道s的上行信干噪比（Signal to Interference plus Noise Ratio, SINR）表达式为：

$$\gamma^{\mathrm{I}}_{m,s} = \frac{\alpha_{m,s} P^{\mathrm{I}}_m h^{\mathrm{I}}_{m,s}}{\sigma^2 + \sum_{n\in\mathcal{N}} \beta_{n,s} P^{\mathrm{V}}_n h^{\mathrm{V}}_{n,s}} \tag{3.1}$$

其中，P^{I}_m和P^{V}_n分别表示V2I-VUE m及V2V-VUE n的发射功率，σ^2表示噪声功率。同理可得V2V-VUE n在子信道 s上到接收机 l的SINR表达式为：

$$\gamma^{\mathrm{V}}_{n,l,s} = \frac{\beta_{n,s} P^{\mathrm{V}}_n h^{\mathrm{V}}_{n,l,s}}{\sigma^2 + \sum_{m\in\mathcal{M}} \alpha_{m,s} P^{\mathrm{I}}_m h^{\mathrm{I}}_{m,l,s} + \sum_{j\in\mathcal{N}, j\neq n} \beta_{j,s} P^{\mathrm{V}}_j h^{\mathrm{V}}_{j,l,s}} \tag{3.2}$$

于是V2I-VUE m 和V2V-VUE n在时隙 t的相应上行链路吞吐量可以汇总为：

$$R^{\mathrm{I}}_m(t) = \sum_{s\in\mathcal{S}} B \cdot \log_2(1 + \gamma^{\mathrm{I}}_{m,s}(t)), \tag{3.3}$$

$$R^{\mathrm{V}}_{n,l}(t) = \sum_{s\in\mathcal{S}} B \cdot \log_2(1 + \gamma^{\mathrm{V}}_{n,l,s}(t)) \tag{3.4}$$

其中B表示子信道带宽。本章节考虑V2V-VUE处于单播通信模式，且将距离发射机n最近的V2V-VUE l作为接收机，因此下文将$R^{\mathrm{V}}_{n,l}$简写为R^{V}_n。值得注意的是，由于路径损耗取决

于收发机之间的距离，因此相应VUE接收机的SINR分布，以及VUE的吞吐量分布，将随着车辆的移动而变化。

3.1.2 问题建模

本节将根据不同VUE的相应指标及约束建立优化问题。

如前所述，考虑V2I-VUE执行具有高吞吐量要求的上行任务，对传输速率需求较高，而V2V-VUE则周期性分发安全相关信息，其具备低延迟和高可靠性需求。于是，对于V2I-VUE具有如下QoS约束：

$$R_m^{\mathrm{I}}(t) \geq R_{\min}^{\mathrm{I}} \tag{3.5}$$

其中$R_{\min}^{\mathrm{I}}$表示V2I-VUE的最低传输速率要求。

可将V2V-VUE的时延可靠性要求建模为在有限的时间预算$T_{\max}$内成功交付大小为L的数据包，因此，在平均意义上，如果以下约束条件成立，则V2V-VUE的时延和可靠性要求将得到满足:

$$R_n^{\mathrm{V}}(t) \geq \frac{L_n(t)}{T_{\max} - (t \bmod T_{\max})} \tag{3.6}$$

其中$L_n(t)$表示在时刻t V2V-VUE n剩余待传输的数据包大小，$(t \bmod T_{\max})$运算表示自该次信息产生已经过去了多少时间。

本章节研究问题的最终目标是通过联合优化子信道选择和传输功率控制，使V2I-VUE的总吞吐量最大化，同时满足V2V-VUE的时延和可靠性要求。由于此前假设V2I-VUE已经预分配了固定传输功率及传输子信道，即$\alpha_{m,s}, \forall m \in \mathcal{M}, \forall s \in \mathcal{S}$与$P_m^{\mathrm{I}}(t), \forall m \in \mathcal{M}$给定，本文不对其进行优化。因此可建立如下优化问题：

$$\begin{aligned} &\max_{\mathcal{P},\mathcal{B}} \sum_{m=1}^{M} R_m^{\mathrm{I}}(t) \\ &\text{s.t.} \quad \text{C(1) - C(2):} \quad (3.5), (3.6) \\ &\qquad \text{C(3):} \quad \sum_{s\in\mathcal{S}} \beta_{n,s} \leq 1, \beta_{n,s} \in \{0,1\}, \forall n \in \mathcal{N} \\ &\qquad \text{C(4):} \quad P_n^{\mathrm{V}}(t) \leq P_{\max}^{\mathrm{V}}, \quad \forall n \in \mathcal{N} \end{aligned} \tag{3.7}$$

其中$\mathcal{P} = \{P_n^{\mathrm{V}}(t) | n \in \mathcal{N}\}$和$\mathcal{B} = \{\beta_{n,s} | n \in \mathcal{N}, s \in \mathcal{S}\}$分别表示所有V2V-VUE的传输功率及子信道选择集合。约束条件C(1)和C(2)分别表示V2I-VUE和V2V-VUE的QoS要求。约束条件C(3)表示每个V2V-VUE只能占用一个子信道，约束条件C(4)表示V2V-VUE可以使用的最

大传输功率。

由于式(3.7)中所定义的优化问题是一个混合整数非线性规划（Mixed Integer Nonlinear Programming, MINLP）问题，求解复杂度较高，而全局信息获取所带来的信令开销又进一步限制了传统中心式优化方法的应用，因此本章节专注于设计分布式的频谱接入方法，并且将在第3.3节中与中心式的暴力搜索方法进行比较，以充分验证算法性能。

3.2　基于多智能体强化学习的算法设计

本章节将使用MARL来解决上一节中所给出的问题，本章节首先基于Dec-POMDP框架对上一章节中式(3.7)所定义的优化问题进行重新表述，并提出相应算法进行求解。

3.2.1　Dec-POMDP建模

由第二章可知，从Dec-POMDP的角度建模多智能体协作问题，需要具体定义智能体的环境观测空间、动作空间和奖励函数。图 3.2形象化地展示了该MARL问题的交互框架。接下来将具体进行介绍。

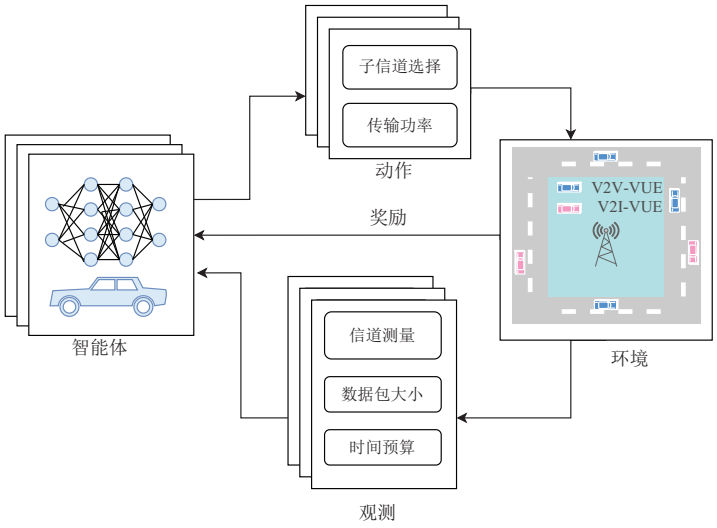


图 3.2　MARL框架示意图

观测空间：文章[20,21,28,29]中均将CSI信息包含在智能体的观测空间内，然而实际情况下在车联网中获得完美的CSI通常较为困难，并且获取CSI所需要的信道估计及反馈等流程也将造成额外的信令开销。另一方面，在现有的3GPP标准车联网资源分配协议中，如LTE Mode 4和NR Mode 2，用户可通过解码控制信息和相应的信道测量结

果，特别是Sidelink参考信号接收功率（Sidelink RSRP, SL-RSRP），来进行频谱资源选择。SL-RSRP反映了如果用户选择在相应的资源块上传输数据，将会接收到的干扰功率水平[44]。在这些方案中，信道估计对于用户成功进行信道资源选择是非必要的。综合以上考虑，在此处的观测空间设计中将不包含CSI信息，主要使用干扰功率测量结果作为观测信息来进行频谱接入控制。

观测空间主要包含感知到的信道干扰测量、待传输的数据包大小和时间预算。具体来说，智能体n在时间t接收到的观测$o_n(t)$由以下三部分信息组成：1）上一时刻，智能体n，即V2V-VUE发射机n，在所有子信道s上经历的干扰功率$\mathbb{I}_n = [I_{n,s}]_{s\in\mathcal{S}}$，其中$I_{n,s} = \sum_{m\in\mathcal{M}} \alpha_{m,s} P_m^{\mathrm{I}} h_{m,n,s}^{\mathrm{I}} + \sum_{j\in\mathcal{N},j\neq n} \beta_{j,s} P_j^{\mathrm{V}} h_{j,n,s}^{\mathrm{V}}$；2）剩余待传输数据大小，$L_n$；以及3）剩余传输时间，$T_n$。因此，此处提出的CSI无关（CSI-independent）的状态空间可表述为：

$$o_n(t) = (\mathbb{I}_n, L_n, T_n) \tag{3.8}$$

在每个时刻t，在所有智能体的联合动作应用于环境之后，干扰测量状态转移由所有智能体当前选择的动作、随机信道变化等决定。剩余的待传输数据大小L_n由当前时间相应的传输速率决定，时间预算T_n减少一个时隙。

进一步，此处给出观测空间包含CSI的另一设计，即CSI-involved版本。具体来说，CSI-involved版本观测空间引入了两部分额外的CSI：1）从V2V 发射机n到相应接收机l（即与其最近的相邻车辆）的信道增益$\mathbb{H}_{n,l}^{\mathrm{V}} = [h_{n,l,s}^{\mathrm{V}}]_{s\in\mathcal{S}}$；2）从V2V 发射机$n$到基站的干扰信道增益$\mathbb{H}_n^{\mathrm{V}} = [h_{n,s}^{\mathrm{V}}]_{s\in\mathcal{S}}$。因此，CSI-involved版本观测空间如下所示：

$$\tilde{o}_n(t) = (\mathbb{I}_n, \mathbb{H}_{n,l}^{\mathrm{V}}, \mathbb{H}_n^{\mathrm{V}}, L_n, T_n) \tag{3.9}$$

为了获得式(3.9)中信道增益信息$\mathbb{H}_{n,l}^{\mathrm{V}}$，需要在接收端进行信道估计，然后反馈回发射端。而对BS的干扰信道增益信息$\mathbb{H}_n^{\mathrm{V}}$的获取则需要在BS进行信道估计，然后反馈给V2V-VUE。与式(3.9)相比，式(3.8)中的观测空间设计通过去除CSI信息，避免了信道估计和反馈信道机制。此外对于干扰功率测量$\mathbb{I}_n$的获取，只需要VUE进行物理层功率检测即可，而无需获取具体的信道增益并进行解码（此处假设按照章节1.3中描述的Mode 2模式进行干扰功率检测流程进行），因此式(3.8)中的观测空间设计有利于减少信令开销。此外，由于CSI-independent版本的环境观测信息主要使用上一时刻测量得到的各子信道功率水平以及自身传输状态信息，这些信息是易于获取的，且处理时延可忽略不计，因此不会带来额外性能损失。

动作空间：在该研究问题中，每个V2V-VUE优化子信道选择s和发射功率选择p的联

合动作，所有智能体的动作空间$\mathcal{A}$一致。具体来说，由于V2V-VUE复用V2I-VUE预分配的子信道，因此V2V-VUE的可用子信道集合为$\mathcal{S}$。为符合实际通信系统，此处智能体可选择的传输功率空间$\mathcal{A}_P$经过离散化处理[57]。综上，智能体的动作空间为：

$$a_n(t) = \{(s,p)|s \in \mathcal{S}, p \in \mathcal{A}_P\} \tag{3.10}$$

奖励函数：总的来说，此处根据式(3.7)中建立的优化问题来设计奖励函数。回顾一下，优化目标是在满足V2V-VUE的延迟和可靠性要求的同时，最大化V2I-VUE总吞吐量。因此设计所有智能体共享的奖励函数如下：

$$\begin{aligned} r(t) =& \lambda_1 \sum_{m=1}^{M} R_m^{\mathrm{I}}(t) + \lambda_2 \sum_{m=1}^{M} F\left(R_m^{\mathrm{I}}(t) - R_{\min}^{\mathrm{I}}\right) \\ &+ \lambda_3 \sum_{n=1}^{N} G_n(t) + \lambda_4 \sum_{n=1}^{N} F\left(R_n^{\mathrm{V}}(t) - \frac{L_n}{T_n}\right) \end{aligned} \tag{3.11}$$

其中$\lambda_i, i \in \{1,2,3,4\}$表示权重。分段函数

$$F(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & x < 0 \end{cases} \tag{3.12}$$

表示对满足式(3.5)和(3.6)中相应VUE的QoS约束的激励。类似的，分段函数

$$G_n(t) = \begin{cases} R_n^{\mathrm{V}}(t), & L_n > 0 \\ c, & L_n \leq 0 \end{cases} \tag{3.13}$$

鼓励V2V-VUE尽早完成数据的传输，其中c是一个比V2V-VUE可达传输速率更大的超参数。此外，式(3.11)中的第一项对应于优化问题(3.7)中的最大化V2I-VUE总吞吐量的目标。由于VUE的传输速率的分布是随着车辆的移动性而变化的，所以式(3.11)中定义的奖励的值分布也将变化。

3.2.2 算法设计

此处首先对将MARL应用于车联网频谱资源分配问题时存在的问题进行总结：1）快速变化的信道和环境部分可观测特性使得智能体需要先进的DRL技术来学习有效的状态表征及动作策略；2）多智能体分布式并发训练导致的非平稳性会阻碍训练过程并降低算法性能；3）环境动态特性变化导致智能体不能准确评估训练效果。本章节接下来将详细介绍为解决这些问题而采取的具体措施。

在DRL中，DNN通常被用作评估Q值的函数近似器，即$Q^{\theta}(s,a)$，其中θ表示DNN的参数。更进一步，为应对环境的部分可观测特性，DNN可以采用循环神经网络（Recurrent Neural Network, RNN）作为隐藏层，利用RNN隐藏层来为智能体维持内部隐藏状态，自动对过去的环境观察结果进行汇总，根据过去获得的部分观测结果来估计环境的全局状态，以此来有效地应对环境仅部分可见的问题[58]。此外，RNN最为人熟知的便是其处理时序问题的能力，其时序数据预测能力能够帮助提取车联网信道的时变特征，使智能体能够学习到有效的状态表征及动作策略。具体而言，此处使用了著名的门控循环单元（Gated Recurrent Unit, GRU）来作为DNN的隐藏层[59]。此外，还结合了Dueling DQN算法[60]为DNN引入对抗结构，通过用不同网络分支分别表示状态价值和动作优势值来实现泛化评估不同动作价值的目的，该结构对应于如下动作价值的分解计算：

$$Q^{\theta}(s,a)=V^{\eta}\left(E^{\xi}(s)\right)+A^{\psi}\left(E^{\xi}(s),a\right)-\frac{\sum_{a'}A^{\psi}\left(E^{\xi}(s),a'\right)}{N_{\text{actions}}} \tag{3.14}$$

其中ξ、η和ψ分别表示共享编码器E^{ξ}、状态价值分支V^{η}和动作优势价值分支A^{ψ}的网络参数[60]。图 3.3展示了本章节中所采用的网络结构示意图。

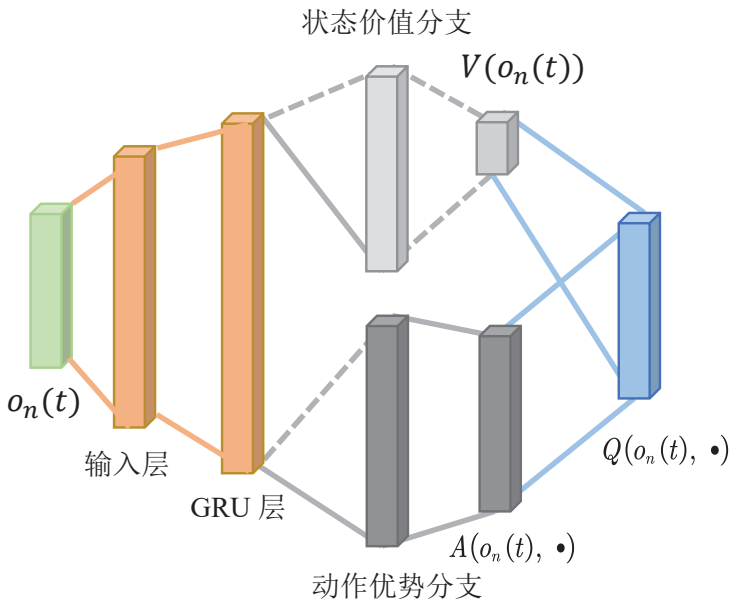


图 3.3　DNN结构示意图

此外，经典的DQN算法面临着价值过估计的问题。此处引入Double DQN算法，通过将动作选择与价值评估相解耦，来解决该问题[61]。具体而言，训练过程中，首先从经验重

放缓存D中均匀采样一批经验样本(s,a,r,s')，然后计算如下损失函数：

$$L(\theta)=\mathop{\mathbb{E}}_{(s,a,r,s')\sim \mathrm{U}(D)}\left[\left(r+\gamma Q^{\theta^{-}}\left(s',\ \mathop{\mathrm{argmax}}_{a'}Q^{\theta}\left(s',a'\right)\right)-Q^{\theta}(s,a)\right)^{2}\right] \tag{3.15}$$

其中θ^{-}表示静态目标网络的参数。令$y=r+\gamma Q^{\theta^{-}}(s',\ \mathop{\mathrm{argmax}}_{a'}Q^{\theta}(s',a'))$表示目标值，通过使用评估网络$\theta$来选取动作，使用目标网络$\theta^{-}$来计算Q值，实现值估计的解耦，有效避免价值过估计问题。此外定义$y-Q^{\theta}(s,a)$为时间差分误差（TD-error）δ。

综上所述，将本章节采用的DQN改进算法合称为D3RQN(Double Dueling Deep Recurrent Q-Network)算法。

该算法遵循分布式训练范式，其也被称为独立学习者（Independent Learner, IL）范式，在训练过程中每个智能体都将其它智能体视为环境的一部分。然而，由于其它智能体在训练探索阶段的动作不可预测，IL训练范式会受到非平稳性的影响。为此，本章节引入了滞后Q-learning来解决这个问题。具体而言，在训练过程中以两种不同的学习率α和β来分别更新Q值估计，其中$0<\beta<\alpha<1$，分别用于乐观估计和悲观估计的TD-误差δ[26]，如下所示：

$$Q(s,a)\leftarrow\begin{cases}Q(s,a)+\beta\delta & \text{if } \delta\leq 0\\ Q(s,a)+\alpha\delta & \text{otherwise}\end{cases} \tag{3.16}$$

在实践中，通常将较大的学习率α固定下来，将较小的学习率放缩为$\beta\cdot\alpha$，随着训练过程的推进，β逐渐增长至1，即与学习率α保持一致，通过这样的调节使智能体在训练的早期阶段可以保持乐观，以对抗那些由于其它智能体不可预测的探索行为导致的负面训练样本，而在训练后期能够逐渐调节实现Q值的准确评估。

为了进一步解决多智能体同时学习的非平稳性问题，本章节还引入了一种经验重放缓冲机制的分布式扩展版本，名为并发经验回放轨迹（Concurrent Experience Replay Trajectories, CERT），其结构如下页图 3.4所示[26]。在训练回合e期间，每个智能体n在时间步t收集经验元组$(o_n(t),a_n(t),r(t),o_n(t+1))$，因为DNN中引入了RNN隐藏层，其训练需要序列状样本，因此此处经验样本以时间序列方式存储（沿图 3.4的时间轴t，每个彩色立方体代表一个经验元组）。在训练过程中，所有的智能体都同时存储经验（如图 3.4所示分别沿着回合数轴e和智能体编号轴n），当需要进行训练时，对所有智能体的序列经验进行同步批采样，并结合式(3.15)和(3.16)进行网络参数更新。该CERT机制可以实现分布式存储，在启动分布式训练过程之前，只需使所有智能体的随机数种子达成一致即可保证训练样本是同步采样的。

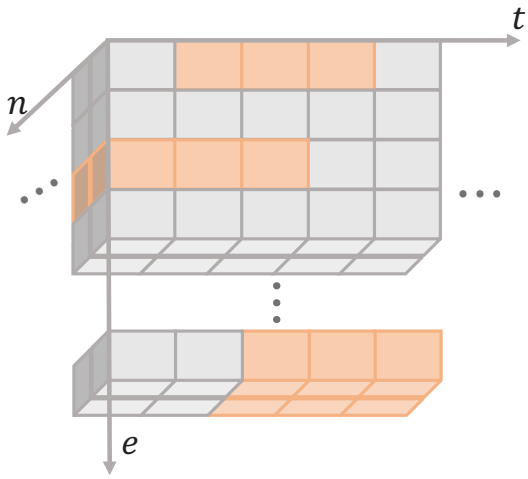


图 3.4 并发经验回放轨迹示意图

许多传统的RL方法专为静态环境而设计，然而，在所研究的C-V2X频谱接入问题中，环境动态特性分布会发生变化。具体来说，式(3.11)中设计的奖励值分布会随着车辆的移动而波动。例如，如果收发机之间的距离变近，使得信道质量变好，该通信链路自然就更容易完成数据包传输，这种指标的提升是由物理世界的内在属性变化造成的，而与具体的频谱接入算法无关。由环境动态特性变化造成的奖励估计偏差可能降低算法性能。为了缓解这个问题，此处引入了近似遗憾奖励（Approximate Regretted Reward, ARR）机制[24]。具体来说，ARR机制借鉴了多臂赌博机（Muti-armed Bandit, MAB）的思想，定义T步交互后的遗憾值ρ为最优策略获得奖励总和与实际策略获得奖励之间的差值期望：$\rho = T\mu^{\star} - \sum_{t=1}^{T} r_t$，其中$\mu^{\star}$表示最大奖励的均值，该遗憾值$\rho$反映了当前策略与最优策略之间的差距，可以用来辅助准确地评估动态环境下的训练性能。尽管在本章节所研究的问题中并没有明确的最优策略，但是一般的静态启发式策略的性能也可以作为参考基准，其差值隐含地反映了潜在的环境变化趋势，因此能够减少奖励估计的波动。在实际训练过程中，需要同时维持两个策略：一个是本工作所提出的基于MARL的实际需要策略，另一个是作为基准的静态启发式策略。将两者获得的式(3.11)中所定义的奖励之差视为智能体实际获得的奖励。在具体实现中，考虑到计算复杂度对训练效率至关重要，此处实际采用了两种简单的启发式算法：随机策略（Random）和轮盘策略（Round-Robin）。此外，值得注意的是，执行ARR技术只会在训练阶段引入额外的计算复杂度，在实际执行阶段没有任何额外开销。

算法 3.1 D3RQN的分布式滞后训练

输入： 学习率α，滞后率β，探索率ϵ，折扣因子γ，样本批量大小N_{B}，采样轨迹长度L_{T}，目标网络更新频率N_{U}

输出： 训练完成的Q网络$\theta_n, n \in \mathcal{N}$

1: 对每个智能体n随机初始化Q网络参数为θ_n，以及相应复制目标网络参数为$\theta_n^- = \theta_n$；
2: **for** 训练中的每一幕轨迹e **do**
3: **for** 每一时间步t **do**
4: **for** 每一个V2V-VUE 智能体n **do**
5: 获取对环境的观测结果$o_n(t)$；
6: 根据ϵ-greedy规则选取动作$a_n(t)$；
7: **end for**
8: 所有智能体执行动作，环境反馈全局奖励r_t；
9: 计算基准奖励r_t'；
10: 计算遗憾奖励$\tilde{r}_t = r_t - r_t'$；
11: **for** 每一个V2V-VUE智能体n **do**
12: 获取新的环境观测$o_n(t+1)$;
13: 将状态转移元组$(o_n(t), a_n(t), \tilde{r}_t, o_n(t+1))$存储进CERT缓存中；
14: **end for**
15: **for** 每一个V2V-VUE智能体n **do**
16: 从CERT中采样得到一批经验样本E_n，其中样本数量为N_{B}，轨迹长度为L_{T}；
17: **for** L_{T}中的每一时间步i **do**
18: **for** $E_{n,i}$中的每一个转移样本$e = (o, a, r, o')$ **do**
19: 计算目标值: $y_e = r + \gamma Q^{\theta_n^-}(o', \underset{a'}{\operatorname{argmax}} Q^{\theta_n}(o', a'))$;
20: 计算TD-error: $\delta_e = y_e - Q^{\theta_n}(o, a)$;
21: 滞后更新： $\hat{\delta}_e = \max\{\delta_e, \quad \beta \cdot \delta_e\}$;
22: **end for**
23: 更新网络参数： $\theta_n \Leftarrow \theta_n + \frac{\alpha}{N_{\mathrm{B}}} \sum_e \hat{\delta}_e \nabla Q^{\theta_n}(o, a)$;
24: **end for**
25: 每N_{U}步更新目标网络参数： $\theta_n^- \Longleftarrow \theta_n$；
26: **end for**
27: **end for**
28: **end for**

$$a_n(t) = \begin{cases} \text{random action}, & \text{with probability } \epsilon \\ \arg\max_a Q^{\theta_n}(o_n(t), a), & \text{with probability} 1-\epsilon \end{cases} \tag{3.17}$$

本工作所提出训练算法总结为算法 3.1。具体来说，算法 3.1的第4至10行描述了智能体与环境的互动过程。在每个时间步t中，每个智能体n根据式(3.17)表示的ϵ-greedy策略选择动作$a_n(t)$，以平衡探索和利用，并获得近似的遗憾奖励。第11行到第14行描述了将经验存储到CERT的过程。第15行至第26行描述了模型更新的实际过程。首先，从CERT中采样一批经验样本。由于该算法的网络结构采用了RNN结构，因此需要执行第17行所描述的顺

序训练。第19行至第23行描述了Double-DQN更新和滞后学习。最后，第25行表示静态目标网络以较低的频率更新，以使评估网络的更新目标保持稳定。

3.3 仿真结果

本章节给出仿真结果以验证此前所提出算法的性能。此处将首先介绍基本的仿真设置，随后验证本章节所提出算法的有效性，并探究状态空间设计对算法性能的影响。为验证算法的稳健性及扩展性，本节还将测试该算法在车辆不同移动速度下的泛化性能，以及在智能体数量增长时的可扩展性。最后，本章节还将进行消融实验，以验证算法不同组件的贡献。

3.3.1 仿真设置

本章节中仿真设置与文章[28]基本保持一致，其中详细定义了车辆移动模型，以及V2I-VUE和V2V-VUE的信道模型。道路拓扑结构如图 3.1所示，仿真区域设置遵循3GPP TR36.885文档中定义的城市道路场景，由四个街区组成，区块宽度和高度分别为250米和433米[62]。与文章[28]类似，为便于仿真处理，此处将仿真区域等比例缩小一半。初始化时，车辆速度在一定范围内随机设置，之后在训练过程中保持匀速运动。车辆根据道路拓扑结构移动，当其到达道路交叉口时，以等概率选择直行或转向。信道模型包括路径损耗、阴影和小尺度瑞利衰落。大尺度衰落和小尺度衰落分别以100ms和2ms的周期更新。表 3.2指定了V2I和V2V链路的信道模型。

表 3.2 信道模型

参数	V2I链路	V2V链路
路径损耗模型	$128.1 + 37.6 \log_{10} d$, d in km	WINNER + B1 Manhattan 视距模型[63]
阴影分布	对数正态分布	对数正态分布
阴影分布标准差	8 dB	3 dB
解相关距离	50 m	10 m
小尺度衰落	瑞利衰落	瑞利衰落
路径损耗和阴影衰落更新间隔	100 ms	100 ms
小尺度衰落更新间隔	2 ms	2 ms

本章节中仿真保持与文章[28]中一样的假设：V2I-VUE的数量以及V2V-VUE的数量（智能体的数量），等于子信道的数量，即$N = M = S$。当子信道和智能体的数量增加时，智能体的联合动作空间的维度会呈指数级增长，从而对算法的可扩展性提出挑战，本节后

续仿真将对此进行研究。为与文章[28]保持一致，本章节仿真设置中，设传输功率空间包含低、中、高三个选项，分别为5、15及23dBm。主要仿真参数见表 3.3，除非特别说明，本节所有的仿真参数都默认设置为表 3.2和 3.3中的数值。

表 3.3 仿真参数设置

参数	值
载波频率	2 GHz
子信道带宽	1 MHz
基站天线高度	25 m
基站天线增益	8 dBi
基站接收机噪声系数	5 dB
车辆天线高度	1.5 m
车辆天线增益	3 dBi
车辆接收机噪声系数	9 dB
车辆速度	$[10, 15]$ m/s
子信道个数 S	$\{4, 8\}$
V2I发射机功率 $P_m^{\mathrm{I}}, m \in \mathcal{M}$	23 dBm
V2V发射机功率$P_n^{\mathrm{V}}, n \in \mathcal{N}$	$\{-100, 5, 15, 23\}$dBm
噪声功率 σ^2	-114 dBm
V2I最小吞吐量$R_{\min}^{\mathrm{I}}$	5 Mbps
V2V时延约束$T_{\max}$	100 ms
V2V数据包大小L	$\{1, 2, \ldots, 6\} \times 1060$ bytes

训练过程中采用的超参数如表 3.4所示。在训练过程中，探索率ϵ逐渐降低以平衡探索和利用。由于在训练后期，当每个智能体都学习到了较好的策略后，评估的准确性变得更加关键，因此滞后学习率β逐渐增加以平衡正负样本之间的更新。在执行阶段，每个智能体感知对环境的局部观测结果，并根据各自训练得到模型选择具有最大Q值的动作。训练期间用于计算共同奖励的中央控制器将不再需要，因此执行阶段是完全分布式的。此外，与文章[28]类似，在训练阶段V2V-VUE的传输数据包大小保持固定，而在执行阶段变化，以验证算法的稳健性。具体实践中，此处选择了对于V2V-VUE而言最难的一种配置，即最大的数据包大小$L = 6 \times 1060$字节，进行训练。此外，为了获得更多样的训练样本，训练过程中会定期重新初始化车辆的位置。本章节仿真代码已开源于Github，具体见[64]。

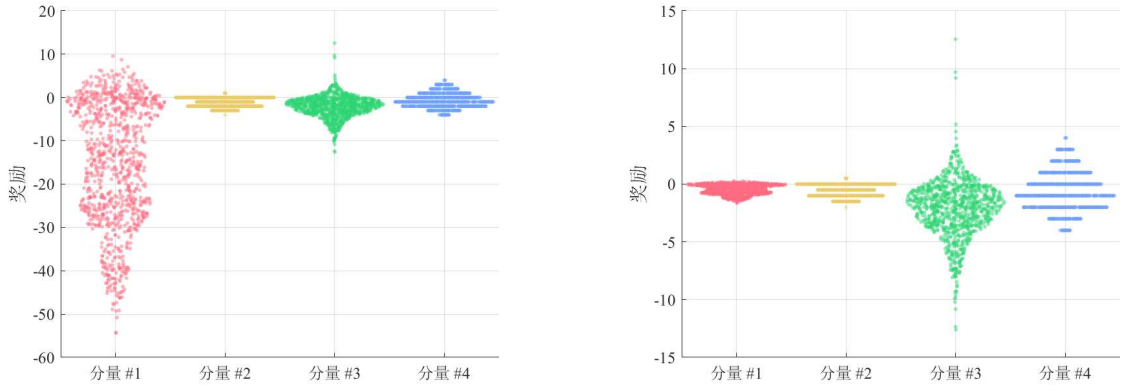
3.3.2 训练超参数选择

通常训练超参数的选择将会很大程度上影响深度学习算法的最终效果，此处也不例外。由于本章所提算法的训练过程中涉及许多超参数（见表3.4），因此很难断言此处所列出的超参数组合即是最佳选择。总体来说，本章节所采用部分超参数，如学习率α、折扣

表 3.4　训练超参数设置

参数	值
学习率α	0.0001
折扣率γ	0.95
探索率ϵ	$1.0 \rightarrow 0.1$
滞后学习率β	$0.2 \rightarrow 0.8$
总探索回合数	15000
总训练回合数	20000
轨迹长度L_{T}	20
批大小N_{B}	32
目标网络更新频率N_{U}	4
CERT大小	1000
奖励权重$\{\lambda_i, i \in \{1,2,3,4\}\}$	$\{0.03, 0.5, 1, 1\}$
激励常数c	1

系数γ、探索率ϵ，是根据经验调整而选择的。例如，折扣系数γ反映了对未来回报的重视程度，这意味着如果γ的值被设置为接近1，算法将更关注长期回报。其他超参数的选择主要是由于硬件限制。例如，批量大小N_{B}取决于计算机内存的大小和可以接受的训练速度。



(a)　　(b)

图 3.5　奖励分量分布：(a) 原始尺度，(b) 加权后尺度

此外，经测试发现奖励权重$\{\lambda_i\}, i \in \{1,2,3,4\}$对算法性能有较大影响，在实际训练过程中这些参数将根据每个奖励分量的大小尺度进行调整。图 3.5中呈现了智能体在训练过程最后10个回合中获得的各奖励分量的值（减去参考基线，即Round-Robin方法）分布情况。图中分量 #1、分量 #2、分量 #3和分量 #4分别对应于奖励函数式（3.11）中V2I-VUE的总吞吐量、V2I-VUE不满足最小吞吐量要求的惩罚、V2V-VUE的总吞吐量，以及V2V-VUE的延迟和可靠性要求。由于本章节中假设V2I-VUE已经预先分配了正交子信道，并且总是以最大功率发射，以及其它的一些仿真设置，如V2I-VUE具有更高的天线增益及更低的噪声系数，V2I-VUE的通信链路要优于V2V-VUE。因此，式（3.11）

中V2I-VUE相应奖励分量的期望值较高。从图 3.5(a)中可以观察到奖励分量 #1的原始未加权幅度比其他奖励分量的波动范围更大。因此，为了平衡V2I-VUE和V2V-VUE的奖励分量，在参数设置中给奖励分量 #1分配了较小的权重。图 3.5(b)中展示了加权后各奖励分量的波动幅度，可以观察到，加权后各分量的波动范围较为一致。

3.3.3　性能验证

仿真中分别将Random方法和Round-Robin方法作为参考基准以计算ARR。具体来说，在Round-Robin方法中，传输功率固定为最大值，子信道选择进行轮转；而在Random方法中，子信道和传输功率都是随机选择。此外，下文还展示了不包括ARR技术的算法性能，即在没有任何参考基准的情况下直接计算式（3.11）中所定义的奖励。

图 3.6展示了训练过程中智能体累积奖励的变化曲线（分别以训练期间获得的相应最大值进行归一化）。从图 3.6中可以看到，随着训练回合数增加，基于两种启发式算法作为参考基准的算法曲线都近似收敛，在训练回合数达到两万时累计奖励基本稳定，而在没有ARR的情况下，前文所提出的算法收敛趋势不明显，这表明了ARR在处理环境动态变化的RL问题上的有效性。此外，还可以看到，采取Round-Robin方法作为参考基准，比采取Random方法获得的奖励要小，但这并不意味着Random方法更适合作为参考基线。出现该现象的原因是因为Round-Robin方法比Random方法作为基准计算式（3.11）中的奖励时，提供了更高的基准值，因此智能体计算实际获得的奖励时要更低一些。

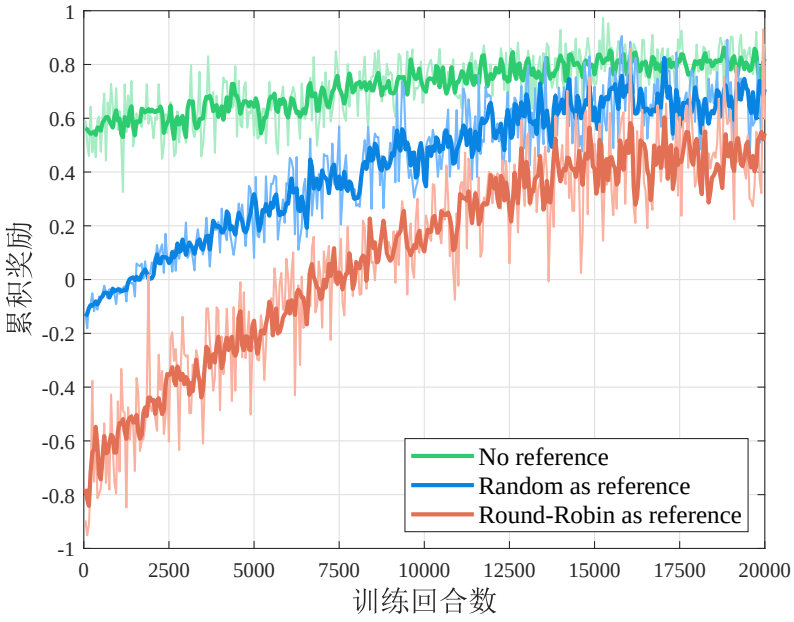


图 3.6　训练曲线（智能体数目为4）

本章节将分别选取若干启发式算法、中心式优化算法以及同样基于MARL的算法进行对比，以验证本章节所提出算法的性能。具体来说，接下来将所提出的算法将分别在没有基线、使用Round-Robin或Random作为基线三种情况下，与中心式优化算法的代表——Brute-force方法（其以最大化V2V-VUE的总速率为目标暴力搜索动作组合，将作为算法性能的上界），两种启发式算法：Round-Robin和Random，以及在文章[28]中所提出的状态空间包含CSI的MARL方法（以下称为*Baseline*）进行对比。此外，3GPP标准协议中的资源分配方案，即NR Mode 2，也加入对比。NR Mode 2是在无基站参与的情况下车辆进行的分布式频谱接入方法，其中发射机车辆使用感知信息，例如参考信号接收功率RSRP测量来选择候选资源。此处选择NR Mode 2的动态模式(以下简称为*Mode 2*)，即每次传输都会选择新的频谱资源，不考虑预留机制[44]。由于强化学习算法的性能会受到环境随机性的较大影响[65]，因此为了验证结果的准确性，以下仿真结果基于20个不同的随机数种子，并以95%的置信区间展示。

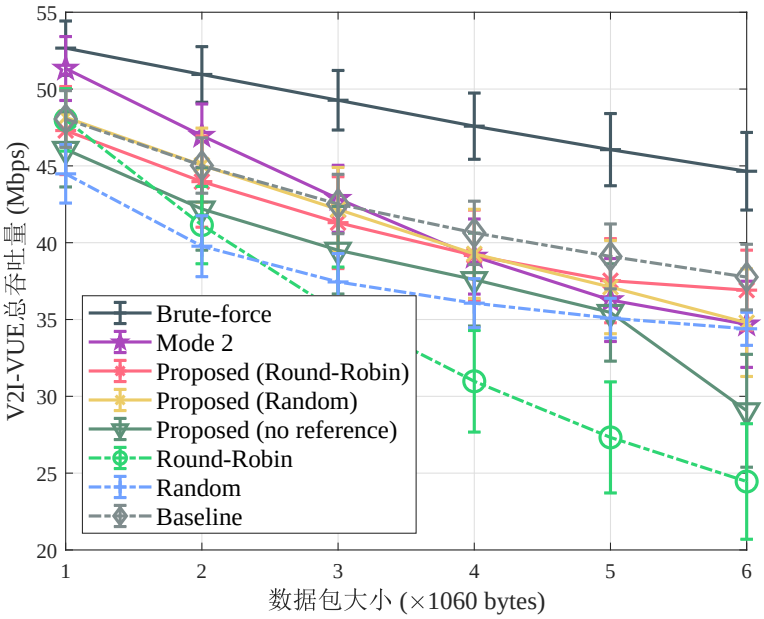


图 3.7　V2I-VUE的总吞吐量（智能体数目为4）

图 3.7展示了V2I-VUE总吞吐量随着V2V-VUE传输数据量大小增加的变化趋势。由于传输数据量增加导致V2V-VUE需要花费更长的时间才能完成传输，为V2I-VUE带来的干扰持续时间也随之增加，因此图 3.7中所有方案的性能都有所下降。即使本章节所提出的CSI-independent版本算法在观测空间中没有使用CSI信息，但从图 3.7中可以看到该算法在使用不同的启发式方法作为基准的情况下（图中的Proposed (Round-Robin)和Proposed (Random)），都表现出比相应的基准方案（Round-Robin和Random）更好的性能，尤其是

当V2V-VUE的传输数据量增加时。此外，还可以从图 3.7中观察到，本工作所提出的算法在去掉ARR技术后（图中的Proposed (no reference)）表现出了更明显的性能下降，说明了ARR技术的有效性。另外一些对比方案，如文章[28]中提出的对比方案（图中Baseline方案）及Mode 2方案，在某些情况下展现了更好的性能。这是由于所有VUE共享频谱资源，V2I-VUE的总吞吐量和V2V-VUE的数据包交付成功率之间存在着权衡关系。结合后续仿真结果，此处微弱的性能劣势将在V2V-VUE的数据包交付成功率指标上实现增益。

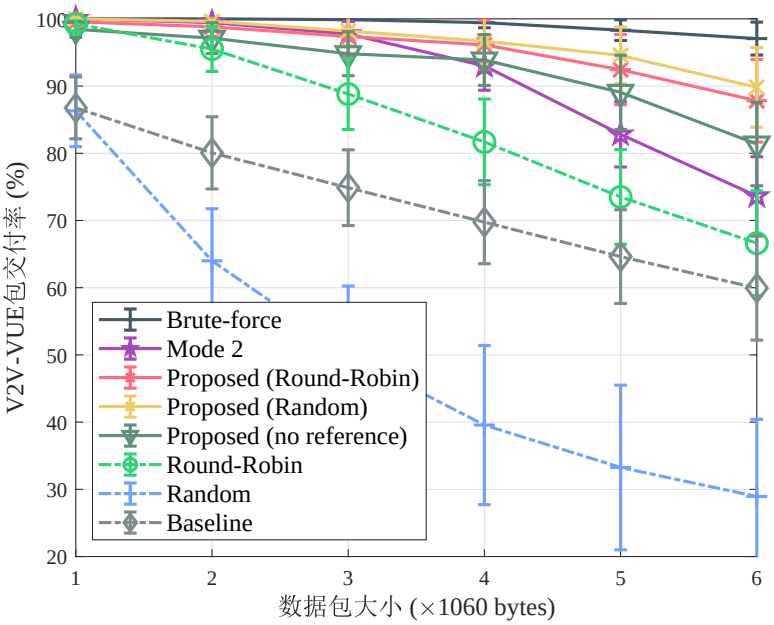


图 3.8　V2V-VUE的数据包交付率（智能体数目为4）

图 3.8展示了V2V-VUE的数据包交付率随着传输数据包大小增加的变化关系。随着传输数据包大小的增加，包括Brute-force在内的所有方案的成功交付率都呈现下降趋势。本工作所提出的分布式CSI-independent版本算法，无论是采用Round-Robin还是Random作为参考基准，甚至去掉ARR技术，均呈现出仅次于接近完美的中心式Brute-force方法的性能。随着传输数据量的增加，所提出的算法呈现出明显优于其它对比方案的性能。此外，从图 3.8中可以观察到，去除ARR技术的算法版本性能下降趋势更明显，这进一步表明了ARR技术采用静态基准策略来辅助跟踪环境动态变化特性的有效性。

本节接下来将进一步评估本章节所提出算法在智能体数量增加时的性能表现。由于仿真设置中遵循了文章[28]中的假设：子信道的数量等于智能体的数量，因此当智能体数目增加时，其动作空间的维度也会随之增加。此外，所有智能体的联合动作空间将随着智能体数目的增加而呈指数级增长，对算法的可扩展性提出了挑战。由于Brute-force方法穷举搜索所耗费时间急剧增长，因此，在接下来智能体数目增加的仿真结果中，不再将其纳入

考量。

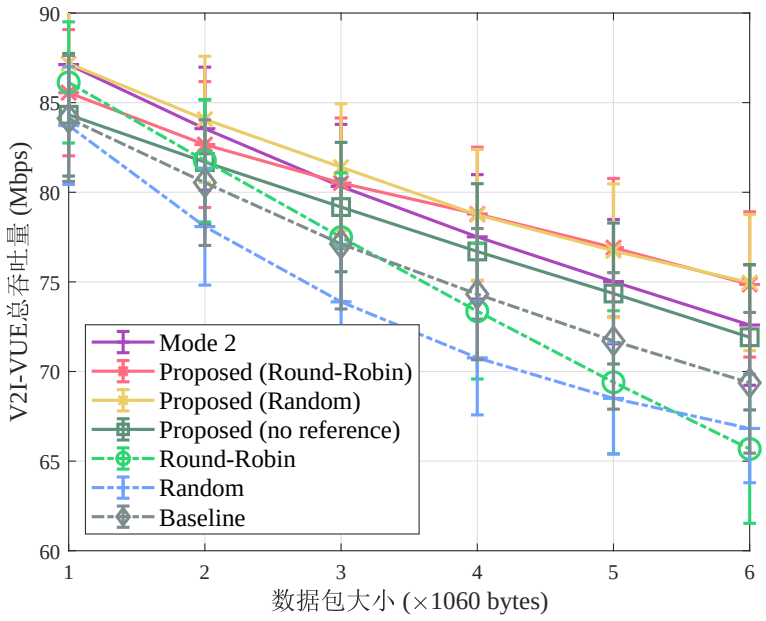


图 3.9　V2I-VUE的总吞吐量（智能体数目为8）

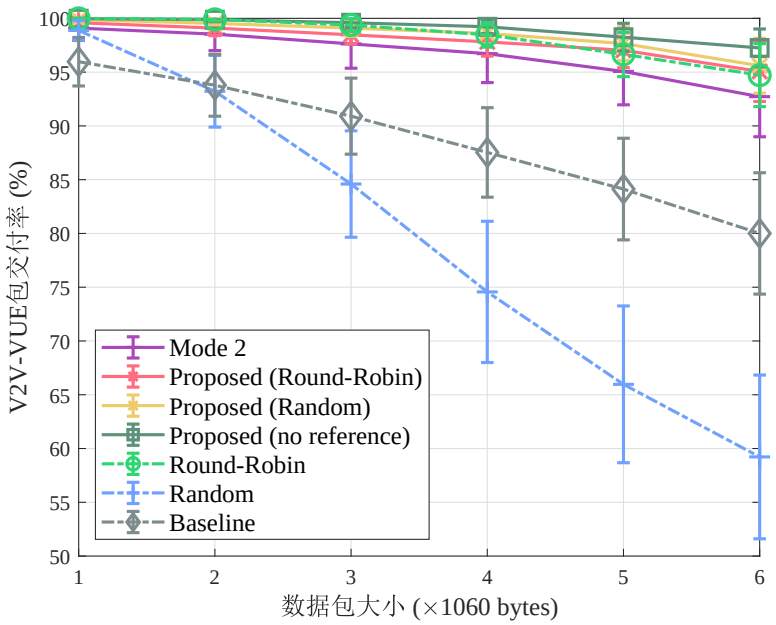


图 3.10　V2V-VUE的数据包交付率（智能体数目为8）

图 3.9展示了V2V-VUE智能体数目增加到8时，V2I-VUE总吞吐量与传输数据包大小的关系。本工作所提出的算法，无论是采取Round-Robin和Random作为基准，还是去除ARR在没有任何基准参考的情况下直接计算奖励的版本，都表现出了与Mode 2相当的性能，并且与其它对比方案相比呈现出明显的性能优势，尤其是当传输数据量增加时。

图 3.10展示了在V2V-VUE智能体数目增加到8时，V2V-VUE的数据包交付率与传输数据量大小之间的关系。从图 3.10中可以观察到，由于收发机之间的距离随着车辆密度的增

加而减小，本工作所提出的CSI-independent版本算法总是呈现出近乎完美的性能，并优于所有其它对比方案。结合图 3.9和 3.10，可以认为本章节所提出的算法可以扩展到更多智能体的情况，并对传输数据量大小的变化具有鲁棒性。

3.3.4　观测空间设计对算法性能影响

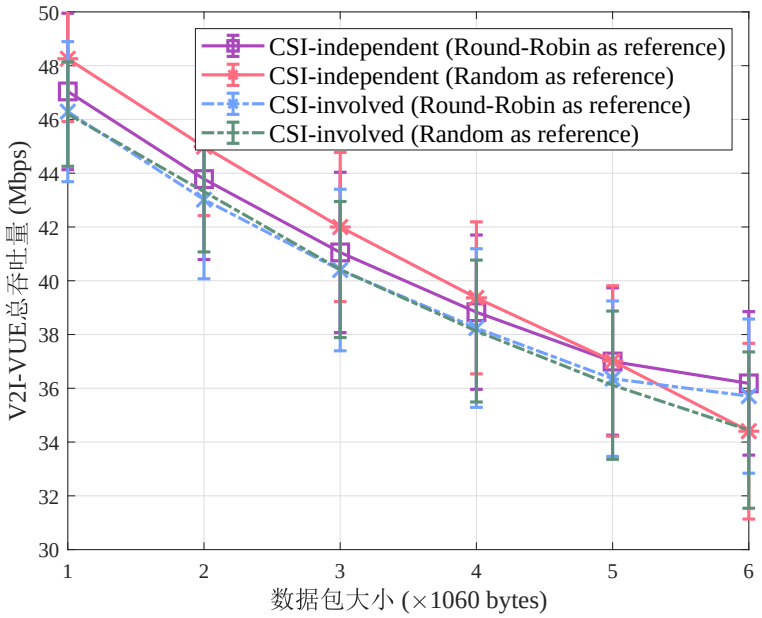


图 3.11　观测空间中是否包含CSI对V2I-VUE总吞吐量指标影响对比

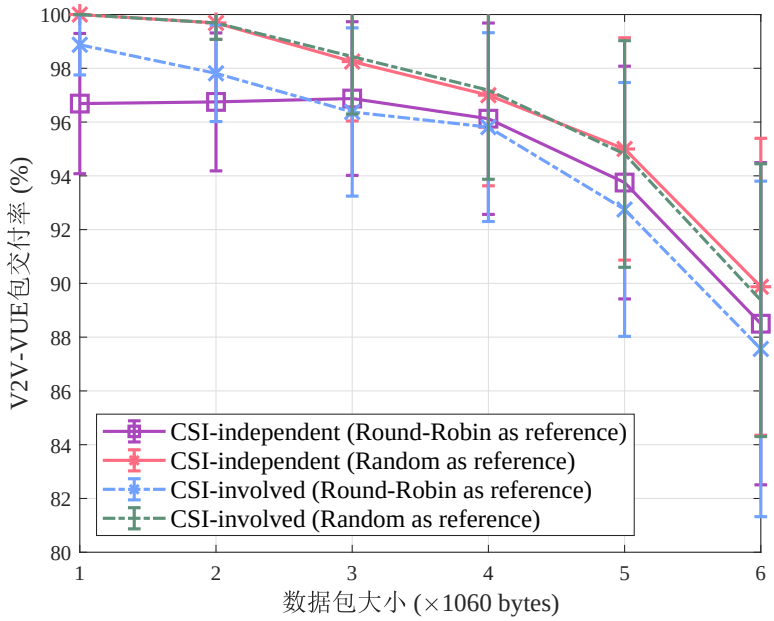


图 3.12　观测空间中是否包含CSI对V2V-VUE包交付率指标影响对比

本章节对CSI-independent及CSI-involved两种不同观测空间设计版本对算法性能的影响进行研究。这两种观测空间设计的唯一区别在于D3RQN的输入是否包含CSI。此处分

别以Round-Robin和Random作为参考，比较了本工作所提算法基于CSI-independent和CSI-involved两种观测空间的性能。具体来说，图 3.11展示了V2I-VUE的总吞吐量指标，而图 3.12展示了V2V-VUE的数据包交付率指标，随着传输数据量大小增加的变化关系。从图 3.11和图 3.12中仅能观察到基于CSI-independent和CSI-involved版本不同观测空间时算法的细微性能差异。就V2I-VUE的总吞吐量指标而言，CSI-independent观测空间版本甚至呈现出轻微的性能优势。该现象说明得益于该工作提出算法中采用的一系列先进DRL技术，即使没有CSI，智能体也能学习到有效的策略。从图 3.7到图 3.10中可以观察到，基于干扰测量而不需要CSI来进行频谱资源选择的Mode 2对比方案也表现出了优秀性能，这也印证了此前的判断。

总的来说，综合考虑V2I-VUE总吞吐量和V2V-VUE包交付率这两个指标，可以得出结论：本章节所提出的算法基于CSI-independent观测空间可以实现与假设了完美CSI反馈的CSI-involved版本相当的性能。

3.3.5 车辆移动速度对算法性能影响

本小节进一步研究了车辆移动速度对本章所提出算法性能的影响。此外，需要注意的是，为了评估该算法的泛化能力，以下不同车辆移动速度下的测试结果均基于以[10,15]m/s速度训练得到的同一模型。此处仿真设置考虑有4个V2V-VUE智能体，数据包大小为6×1060字节，车辆移动速度范围包含四个区间，即[10,15]m/s、[15,20]m/s、[20,25]m/s和[25,30]m/s。从图 3.13和 3.14中可以观察到，随着车辆速度的增加，本章节所提出算法的性能略有下降。这是因为当车辆移动速度增加时，智能体观测到的环境状态变化与在训练阶段（较低移动速度）所观察到的状态相比会更剧烈，因此会产生细微的泛化性能损失，但总体来说，算法的性能损失在可接受范围内，且该损失可通过在训练过程中增加车辆移速范围，使智能体经历更多样的环境条件来应对。

3.3.6 扩展性验证

本小节将从算法性能与智能体数量变化关系的角度来研究算法的可扩展性。图 3.15和图 3.16展示了V2I-VUE的平均吞吐量和V2V-VUE的数据包交付率随着智能体数量增加的变化趋势。可以从图 3.15中观察到，V2I-VUE的平均吞吐量在智能体数量增加时基本保持稳定。这是因为在本章节中假设V2I-VUE被预先分配了正交子信道，而且仿真中假设其均匀分布，因此V2I-VUE与BS的平均距离基本保持恒定，所以V2I-VUE的平均吞吐量不会受到智能体数量变化的显著影响。

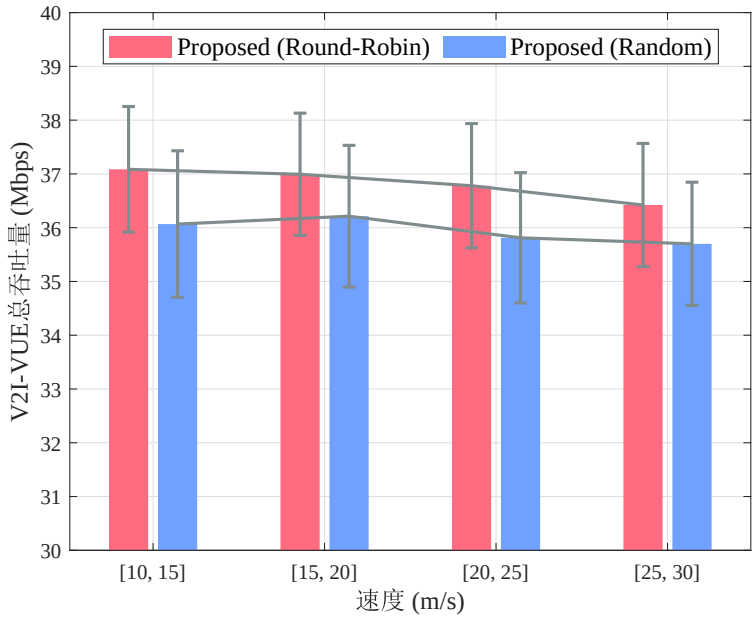


图 3.13　车辆移动速度对V2I-VUE总吞吐量指标的影响

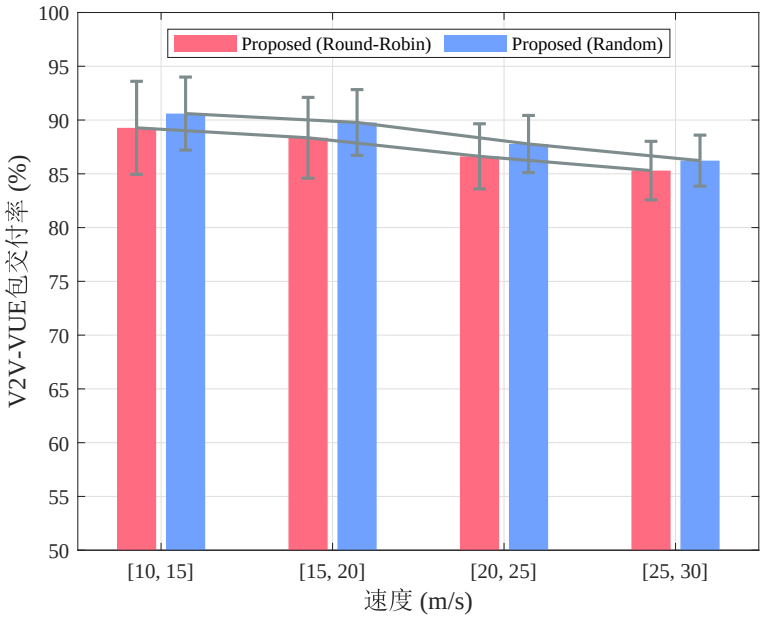


图 3.14　车辆移动速度对V2V-VUE包交付率指标的影响

此外，可以从图 3.16中观察到，随着智能体数量的增加，V2V-VUE的数据包交付率呈现增长趋势。出现该现象是由于在仿真设置中保持了和文章[28]一样的假设，即频谱资源块的数量和智能体数量保持一致，当车辆数量增加时，车辆密度增加，收发机距离减少，而在这种仿真设置下可使用的频谱资源却没有相对减少，因此使得V2V-VUE的SINR相对升高，其性能自然更好。根据图 3.15和图 3.16，以及图 3.9和 3.10中呈现出的本章所提出

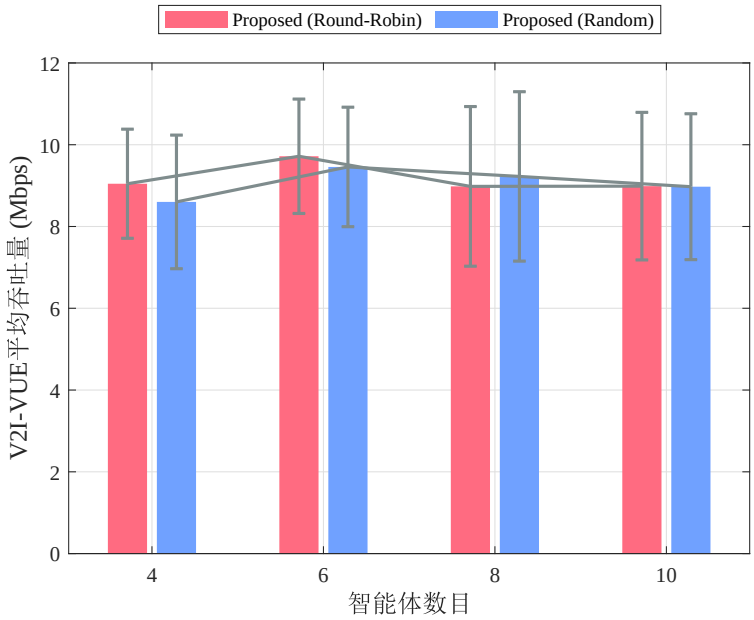


图 3.15　V2I-VUE平均吞吐量随着智能体数目增长变化情况

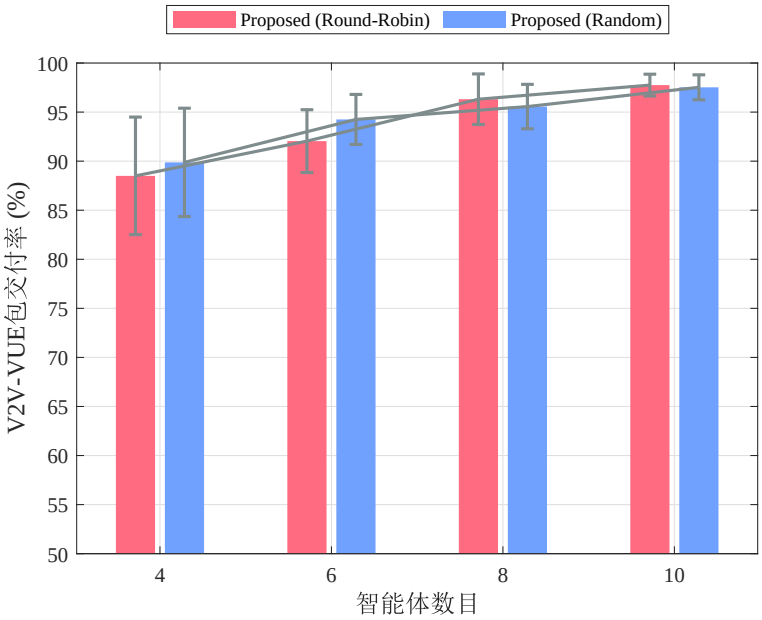


图 3.16　V2V-VUE包交付率随着智能体数目增长变化情况

算法相对其它基准算法的显著优势，此处可以得出结论：本章所提出的算法具备扩展到更多智能体数量的能力。

3.3.7　消融实验及复杂度分析

由于本章节所提出的算法中整合了多种学习技术，因此，为研究各组件各自的贡献，

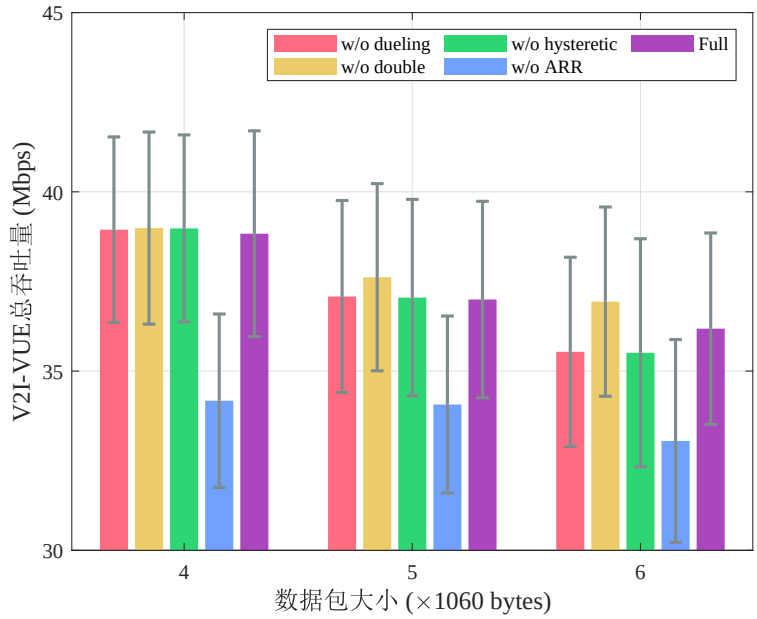


图 3.17　消融实验：V2I-VUE总吞吐量指标

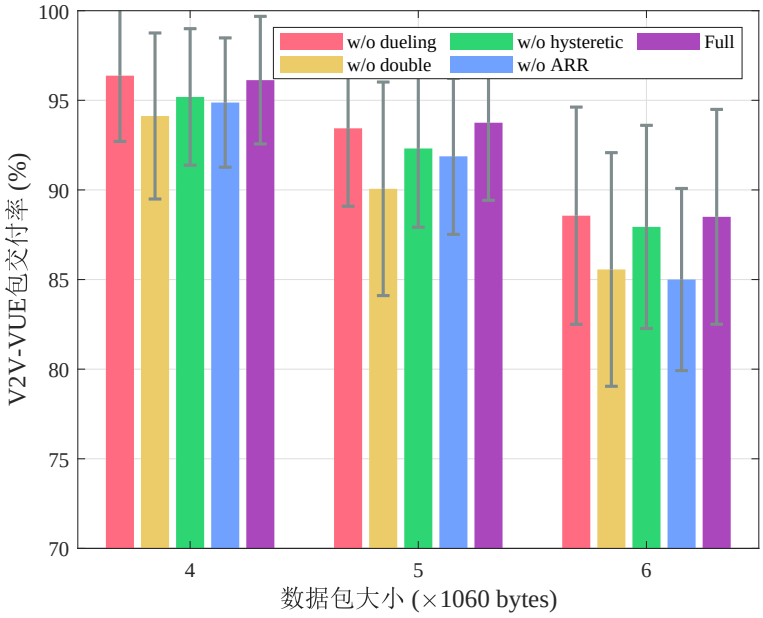


图 3.18　消融实验：V2V-VUE包交付率指标

本小节进行了消融实验。在单独的每个消融项中，从所提算法的完整组合（简称为 *Full*）中分别删除一个组件。具体来说，本部分分别尝试去掉了单独评估状态价值和动作优势值的对抗结构（简称为*w/o dueling*）[60]；解决了值过估计问题的双DQN更新技术（简称为*w/o double*）[61]；解决了多智能体同时更新带来负面影响的滞后学习技术（简称为*w/o hysteretic*）[26]，以及应对环境动态变化改变奖励函数值分布的ARR技术（简称为*w/o ARR*）[24]。由于RNN和CERT技术构成了该算法的基本框架，因此没有对这两种技术进行消融实

验。值得注意的是，如果将上述所有组件都剔除，则会退化为普通的DQN算法。

从图 3.17和图 3.18中可以观察到，本章节所提出的完整版算法在V2I-VUE的总吞吐量和V2V-VUE的数据包交付率这两个指标上表现相对更好。由于V2I-VUE的总吞吐量和V2V-VUE的数据包交付率这两个指标之间存在权衡，因此在某些情况下，完整版本算法可能不如其消融版本。例如，当去除double-DQN组件（图中*w/o double*项）时，该算法在V2I-VUE的总吞吐量指标上表现更好，而在V2V-VUE的数据包交付率指标上不如完整版本。ARR是该算法中最关键的组成部分之一，其能够有效地解决动态环境中学习不稳定的问题，因此，从图 3.17和 3.18中可以观测到去除ARR组件会导致算法性能有明显的下降。其次，去除滞后学习机制也对所提算法的性能有较大影响，这是因为滞后学习有助于协调多智能体的并发学习。此外，去除对抗DQN机制（*w/o dueling*）也会导致算法性能轻微的下降。

最后，对本章节提出算法的计算复杂度进行分析。该算法的计算复杂度实际上取决于所采用的DNN的结构，其大致随着隐藏层的数量和相应的神经元数量的增加而增加。具体来说，在此处的仿真实验中，每个智能体的DNN包括一个有64个神经元的全连接输入层、一个有128个单元的隐藏GRU层和一个有64个神经元的全连接输出层。算法复杂度以Big-O表示法大致为：$O(|S|\cdot|B_0|+3\cdot|B_0|\cdot|Q|+3|Q|^2+3|Q|+|Q|\cdot|B_1|+|B_1|\cdot|A|)$，其中$|S|$表示输入（状态空间）的维度，$|A|$表示输出（动作空间）的维度，$|B_0|$和$|B_1|$表示输入和输出全连接层的神经元数量，$|Q|$表示隐藏门控循环单元（GRU）层的神经元数量。

本章节所提出的算法是用Python和Pytorch[66]实现的。在本文的测试环境中，使用一台处理器为英特尔酷睿i5-8265U的笔记本电脑，智能体每次行动选择大致花费约6.7×10^{-4}秒。如果结合当下众多先进的神经网络加速技术，如模型压缩、量化、GPU，甚至专门的FPGA进行硬件加速，那么执行效率仍有很大的提升空间[67]。

3.4 本章小结

本章节以最大化V2I-VUE的总吞吐量，同时满足V2V-VUE的延迟和可靠性要求为优化目标，提出了一种基于多智能体强化学习的车联网分布式频谱接入算法。通过本章节所提出方法，V2V-VUE可在仅需局部环境观测且无需信道状态信息基础上联合优化子信道和传输功率的选择，并学会在没有智能体间通信机制辅助的情况下进行隐式协作。该算法结合了一系列先进的DQN算法改进，并且为了应对多智能体并发学习所引起的非平稳性，该算法结合了滞后Q-learning和一种名为CERT的分布式重放缓冲改进技术。此外，本章节还

引入了一种近似遗憾奖励机制，来解决环境动态变化导致智能体难以准确评估训练效果的问题。以上所有的算法改进组件为最终完整版算法相对于对比方案呈现出的性能优势做出了各自贡献。此外，由于该算法使用不包含CSI的环境观测信息也能实现与包含CSI版本相当的性能，因此该算法有助于降低信令开销，提高频谱利用率。最后，仿真实验也验证了该算法相对于对比方案可以扩展到更多的智能体。

4 引入智能体间通信机制的分布式频谱接入算法

车联网信道接入问题是一个典型的多智能体系统问题，从单个智能体的角度看，其获得的收益会同时受到环境和其它智能体策略的影响。在基于强化学习设计分布式信道接入算法时，需要考虑不同智能体间交互对训练效果的影响，而允许智能体间进行信息交换是使智能体更好地实现协作的一种有效途径。本章节在第3章所提出的分布式多智能体算法基础上引入了智能体间通信机制，即智能体在每个时间步除了决策子信道选择、传输功率等信道接入直接相关动作外，还需要决策向其它智能体发送的消息内容。本章节所设计算法希望通过信息交互机制来实现更优的多智能体任务协作，在车联网信道接入问题上获得更好的表现。

4.1 系统模型

4.1.1 问题建模

此处考虑与第3章中类似的问题建模，为便于回顾，此处仅列出优化问题，本章中出现的部分符号定义如无特殊说明均可参见表 3.1。本章节所研究问题的优化目标是通过联合优化子信道选择和传输功率控制，使V2I-VUE的总吞吐量最大化，同时满足V2V-VUE的时延可靠性要求，即如下优化问题：

$$
\begin{aligned}
&\max_{\mathcal{P},\mathcal{B}} \sum_{m=1}^{M} R_m^{\mathrm{I}}(t) \\
&\text{s.t.} \quad \text{C(1):} \quad R_m^{\mathrm{I}}(t) \geq R_{\min}^{\mathrm{I}} \\
&\qquad \text{C(1):} \quad R_n^{\mathrm{V}}(t) \geq \frac{L_n(t)}{T_{\max} - (t \bmod T_{\max})} \\
&\qquad \text{C(3):} \quad \sum_{s \in \mathcal{S}} \beta_{n,s} \leq 1, \beta_{n,s} \in \{0,1\}, \forall n \in \mathcal{N} \\
&\qquad \text{C(4):} \quad P_n^{\mathrm{V}}(t) \leq P_{\max}^{\mathrm{V}}, \quad \forall n \in \mathcal{N}
\end{aligned}
\tag{4.1}
$$

其中$R_m^{\mathrm{I}}(t)$和$R_n^{\mathrm{V}}(t)$分别表示V2I-VUE m及V2V-VUE n的传输速率。在此前第3章中，已经基于多智能体强化学习提出了一种完全分布式的频谱接入算法，且仿真结果证明了该算法的有效性。本章将引入智能体间通信机制，通过端到端的训练使智能体学习通信协议，实现更好的协作效果。

4.1.2　Dec-POMDP建模

首先，基于Dec-POMDP对式(4.1)中优化问题进行建模。考虑为智能体间引入了通信机制，因此每个智能体$n \in \mathcal{N}$除了决策子信道和传输功率的联合动作a_n外，还需要决策智能体间进行交互的信息m_n。此处假设该交互信息通过专用的广播控制信道发送给所有智能体，每个智能体将接收到的来自其它智能体的交互信息作为其决策算法的输入，通过信息共享机制，来实现智能体间更好的协作。图4.1展示了引入通信机制的多智能体强化学习交互框架。

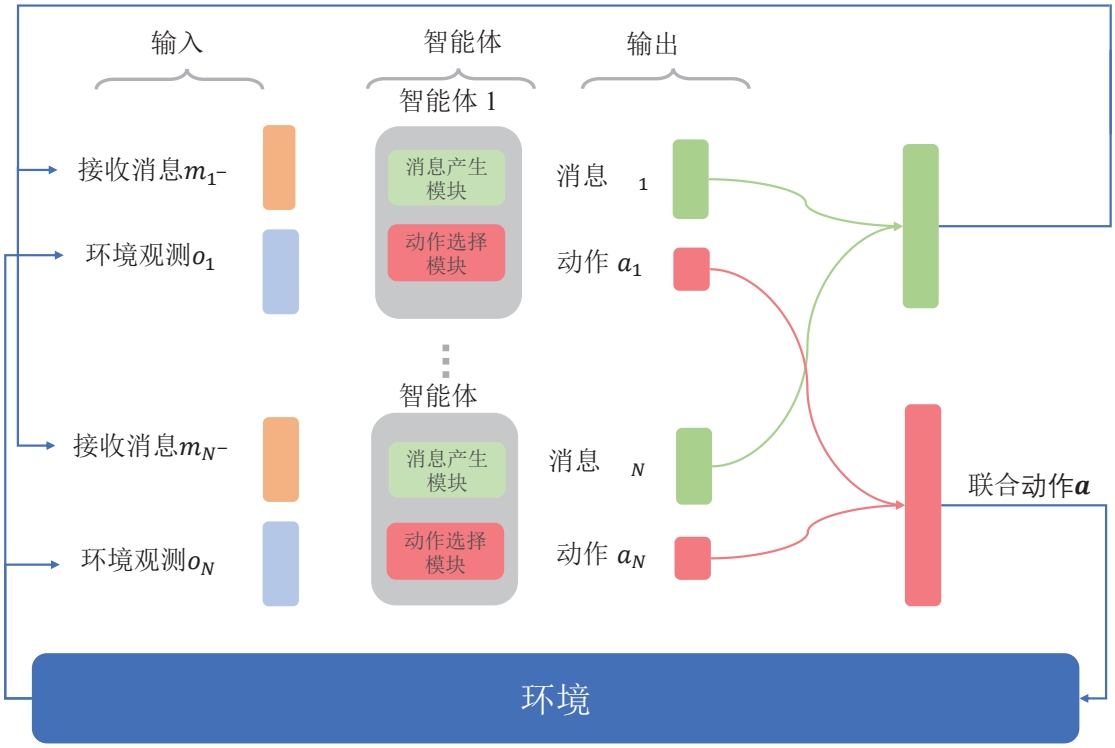


图 4.1　引入通信机制的多智能体强化学习交互框架

对每个V2V-VUE智能体$n \in \mathcal{N}$，其对环境的观测结果由以下部分组成：(1) 上一时刻，其在所有子信道$s \in \mathcal{S}$上经历的干扰功率$\mathbb{I}_n = [I_{n,s}]_{s\in\mathcal{S}}$，其中

$$I_{n,s} = \sum_{m\in\mathcal{M}} \alpha_{m,s} P_m^{\mathrm{I}} h_{m,n,s}^{\mathrm{I}} + \sum_{j\in\mathcal{N},j\neq n} \beta_{j,s} P_j^{\mathrm{V}} h_{j,n,s}^{\mathrm{V}} \tag{4.2}$$

2）当前剩余待传输数据大小，L_n；（3）当前剩余传输时间T_n。因此，智能体的观测空间如下所示：

$$o_n(t) = (\mathbb{I}_n, L_n, T_n) \tag{4.3}$$

由于引入了智能体间通信机制，因此上一时刻其余智能体广播的交互信息也将会作为智能体的决策输入。此外，将每个智能体独有的身份标识作为决策输入的一部分，也有助于各自学习到有区分度的策略；在时序任务中，将智能体过去输出的动作作为新一时刻的输入，有助于智能体学习到时序依赖关系[33]。因此，智能体的输入还将包含智能体序号n以及上一时刻动作输出$a_n(t-1)$。由于智能体决策所需的输入，除了当前时刻的传输状态L_n和T_n需要实时获取外，其余的信息，如上一时刻测量得到的信道干扰功率$\mathbb{I}_n$、上一时刻的动作输出和上一时刻接收到的广播交互信息，均不需要实时获取，因此环境感知不会引入额外的时延。

如前所述，智能体的动作分为频谱接入动作和信息交互动作两类。频谱接入动作a_n，即子信道选择和传输功率选择的联合动作：

$$a_n(t) = \{(s,p)|s \in \mathcal{S}, p \in \mathcal{A}_P\} \tag{4.4}$$

其中$\mathcal{S}$为数据传输的子信道集合，$\mathcal{A}_P$为离散化的传输功率集合。

智能体的信息交互动作决定其每个时刻广播的信息内容，即广播的信号向量m_n。智能体交互的信息量，或者说m_n的维度，取决于广播控制信道的带宽，为符合实际通信系统，此处假设控制信道仅承载离散数据，且每个智能体能使用的广播信道数据带宽为K比特，则控制信息m_n是一个K维的0-1矢量，即：

$$m_n(t) = \{0,1\}^K \tag{4.5}$$

由于本章节研究多智能体协作场景，所以为所有智能体设计统一的奖励函数。奖励函数根据优化问题(4.1)定义，需要同时考虑V2I-VUE的传输速率需求及V2V-VUE的传输可靠性需求，因此由以下四部分组成：1）V2I-VUE的总速率；2）V2I-VUE不满足最低传输速率的惩罚；3）V2V-VUE的总速率；4）V2V-VUE不满足时延可靠性要求的惩罚。此处同样沿用章节3.2.2中提出的ARR技术，即引入一项启发式静态策略，将其奖励作为归一化基准以实现在动态变化环境中更好的训练效果。综合上述需求，奖励函数设计如下：

$$r(t) = \lambda_1 R_1 + \lambda_2 R_2 + \lambda_3 R_3 + \lambda_4 R_4 \tag{4.6}$$

其中，奖励函数中的R_1分量对应于优化目标中的V2I-VUE的总速率要尽可能大这一需求。

令$\tilde{R}_m^{\mathrm{I}}(t)$表示静态基准策略下V2I-VUE m的传输速率，则R_1具体设计如下

$$R_1 = \tanh\left[\frac{a \cdot \frac{1}{M}\left(\sum_{m=1}^{M} R_m^{\mathrm{I}}(t) - \sum_{m=1}^{M} \tilde{R}_m^{\mathrm{I}}(t)\right)}{\sum_{m=1}^{M} \tilde{R}_m^{\mathrm{I}}(t)}\right] \tag{4.7}$$

此处通过先将V2I-VUE平均传输速率使用基准策略的相应速率进行归一化，再经过$\tanh(\cdot)$函数运算，能够使得R_1分量的值处于$(-1,1)$之间，便于调整各分量权重。此外，a为可调放缩系数，用于调整该分量的分布位置，此处通常设为10。

R_2表示对于V2I-VUE不满足最低传输速率要求时所给予的惩罚，具体设计如下：

$$\frac{1}{M}\left[\sum_{m=1}^{M} \mathbb{1}\left(R_m^{\mathrm{I}}(t) \geq R_{\min}^{\mathrm{I}}\right) - \sum_{m=1}^{M} \mathbb{1}\left(\tilde{R}_m^{\mathrm{I}}(t) \geq R_{\min}^{\mathrm{I}}\right)\right] \tag{4.8}$$

其中$\mathbb{1}(\cdot)$判断条件是否成立，如成立返回1，否者返回0。此处同样将与静态基准策略的差值作为实际奖励，这样有助于实现在动态变化环境中更准确的奖励评估。

R_3分量对应V2V-VUE的总速率奖励，类似于R_1分量，此处同样通过静态基准策略对其进行归一化处理。具体设计如下：

$$R_3 = \tanh\left[\frac{a \cdot \frac{1}{N}\left(\sum_{n=1}^{N} G_n(t) - \sum_{n=1}^{N} \tilde{G}_n(t)\right)}{\sum_{n=1}^{N} \tilde{G}_n(t)}\right] \tag{4.9}$$

其中

$$G_n(t) = \begin{cases} R_n^{\mathrm{V}}(t), & L_n > 0 \\ c, & L_n = 0 \end{cases} \tag{4.10}$$

以及

$$\tilde{G}_n(t) = \begin{cases} \tilde{R}_n^{\mathrm{V}}(t), & \tilde{L}_n > 0 \\ c, & \tilde{L}_n = 0 \end{cases} \tag{4.11}$$

分别表示采用本章所提出算法与采用静态基准策略下V2V-VUE n的对应奖励分量。$R_n^{\mathrm{V}}(t)$和$\tilde{R}_n^{\mathrm{V}}(t)$分别表示V2V-VUE n的速率及静态基准策略下V2V-VUE n的相应速率。c为一可调常数，通常设置为比V2V-VUE可达速率略大的值，表示当传输完成时（$L_n = 0$），为其赋予更大的奖励，通过这一设置可鼓励V2V-VUE尽早完成传输。

最后，R_4分量表示对于V2V-VUE不满足时延可靠性约束时所给予的惩罚，具体设计

如下：

$$\frac{1}{N}\left[\sum_{n=1}^{N}\mathbb{1}\left(R_n^{\mathrm{V}}(t)\geq\frac{L_n(t)}{T_{\max}-(t \bmod T_{\max})}\right)-\sum_{n=1}^{N}\mathbb{1}\left(\tilde{R}_n^{\mathrm{V}}(t)\geq\frac{\tilde{L}_n(t)}{T_{\max}-(t \bmod T_{\max})}\right)\right] \tag{4.12}$$

此处从平均意义上刻画V2V-VUE的时延可靠性约束。如果V2V-VUE n要在剩余时间（即$T_{\max}-(t \bmod T_{\max})$）内传输完大小为$L_n(t)$的数据，则当前速率$R_n^{\mathrm{V}}(t)$应至少持平要达成该需求的平均速率。此外，此处同样将与静态基准策略的差值作为实际奖励。

在本章节奖励函数的设计中，通过对各奖励分量进行合理的放缩及归一化，能够有助于后续训练环节超参数的选择。

4.2　引入通信机制辅助智能体协作的算法设计

图 4.1中展示了该引入通信机制的MARL交互框架，由于引入多智能体间通信机制，因此智能体在进行信道接入决策时，不仅可以根据当前自己检测到的信道干扰测量结果、待传输数据量等自身可观测的局部环境状态进行决策，还可以通过接收到的其它智能体广播信息来获取其它智能体的状态、策略等信息，以实现更好的协作效果。具体来说，如图4.2所示，每个智能体包含动作选择模块以及消息产生模块。动作选择模块决策信道接入的相关动作，即子信道选择和传输功率选择；消息产生模块则产生相应的交互信息。两个模块均使用神经网络实现，分别根据接受收到的环境观测及其它智能体发送的交互信息来生成相应的输出。

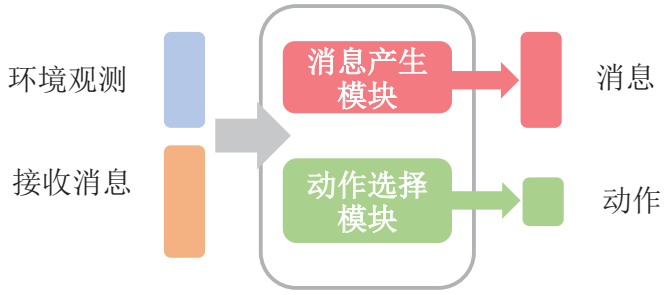


图 4.2　智能体组成结构

4.2.1　动作选择模块

令$\theta_n, n\in\mathcal{N}$表示智能体$n$维护的动作选择模块网络参数，给定当前时刻环境观测$o_n(t)$和上一时刻接收到的别的智能体广播交互信息的连接（Concatenation）：

$$m_{n^-}(t-1)=\bigoplus_{j\neq n, j\in\mathcal{N}} m_j(t-1) \tag{4.13}$$

作为输入，再加上上一时刻该智能体输出的动作$a_n(t-1)$及身份标识n，当前时刻决策动作对应的Q值为$Q^{\theta_n}(o_n(t), m_{n^-}(t-1), a_n(t-1), n)$。智能体采用经典的DQN算法作为基础的动作选择算法，在训练阶段，智能体按照ϵ-greedy策略选择动作来平衡探索和利用，即以ϵ概率随机选择动作，$1-\epsilon$概率选择使Q值最大的动作：

$$a_n(t) = \begin{cases} \arg\max_a \ Q^{\theta_n}(o_n(t), m_{n^-}(t-1), a_n(t-1), n, a), & \text{with probability} \quad 1-\epsilon \\ \text{random action}, & \text{with probability} \quad \epsilon \end{cases} \tag{4.14}$$

而在执行阶段，智能体直接选择使Q值最大的动作以使长期收益最大化。

4.2.2 消息产生模块

消息产生模块同样采用神经网络实现。为充分发挥端到端训练优势，智能体n的消息产生模块在训练阶段的输出$m_n(t)$为连续实值矢量。由于$m_n(t)$将会在不同智能体间传输，因此可以通过梯度反向传播实现跨智能体的端到端联合训练。而在执行阶段，由于实际通信系统中信道带宽有限，为与实际情况相符，$m_n(t)$需要经过离散化处理。为此，本章引入离散正则单元（Discretize/regularize Unit, DRU），DRU在训练阶段将消息产生模块输出的连续实值进行归一化，在本章节中采用Sigmoid函数实现；而在测试执行阶段则需要将$m_n(t)$离散化处理，此处可以简单根据$m_n(t)$各元素的值是否大于0进行0-1编码。假设在测试阶段每个智能体可使用的控制信道带宽为K比特，即可传输K维的0-1矢量，因此消息产生模块的原始输出对应为K维连续实值矢量，经过DRU逐元素离散化处理后即得K比特离散消息。为了最大限度的减少从连续实值映射到离散编码时产生的离散化误差，此处在训练阶段采取了两种措施。首先，为消息产生模块的输出加上高斯白噪声，以限制在训练阶段消息产生模块连续实值输出可表示的信息量。第二，附带噪声的信息将通过一个正则函数，即Sigmoid函数，来将可用于编码信息的范围限制到$(0,1)$。综上所述，DRU具有如下形式[33]：

$$\mathrm{DRU}(m_n) = \begin{cases} \text{Sigmoid}(\mathcal{N}(m_n, \sigma^2)), & \text{if training} \\ \mathbb{1}\{m_n > 0\}, & \text{otherwise} \end{cases} \tag{4.15}$$

其中$\mathcal{N}(m_n, \sigma^2)$即表示在训练过程中为DRU输入$m_n$加上一定的高斯白噪声。

4.2.3 神经网络结构

每个智能体维护的神经网络由输入网络、中间隐藏层及输出网络组成。由于在本文中

动作选择模块和消息产生模块均由神经网络实现，因此为了更有效的联合训练，动作选择模块和消息产生模块可以共享部分网络参数。此处令动作选择模块和消息产生模块共享输入网络及中间隐藏层，仅在输出层网络进行区分，此处用简单的全连接网络实现即可。执行时，智能体获取环境观测信息和广播交互消息，一并输入到智能体维护的整体神经网络中，经过两个模块共享的神经网络处理后，在输出层分别输出不同动作的Q值以及消息比特。因此，最终整体网络的输出维度为信道接入动作维度$|\mathcal{S}||\mathcal{A}_P|$加上交互消息比特数$K$。

智能体神经网络的输入为元组$(o_n(t), m_{n^-}(t-1), a_n(t-1), n)$，由于输入$n$和$a_n(t-1)$为有限集合内的离散标量，因此可借鉴自然语言处理领域中常用的嵌入（Embedding）技术，通过查表（Lookup）操作将离散标量转化为神经网络更适宜的连续实质矢量嵌入表示，且相应的嵌入表示可通过学习得到[33]。输入中的环境观测信息$o_n(t)$以及其它智能体的交互信息$m_{n^-}(t-1)$可通过全连接层（Multi-layer Perceptron, MLP）产生相同尺寸的嵌入表示，最后将上述同尺寸大小的嵌入表示逐元素相加即可得到最终的嵌入表示，如下式所示：

$$z_n(t) = (\mathrm{ObsMLP}(o_n(t)) + \mathrm{MsgMLP}(m_{n^-}(t-1)) + \mathrm{Lookup}(a_n(t-1)) + \mathrm{Lookup}(n)) \tag{4.16}$$

其中$\mathrm{ObsMLP}(\cdot)$和$\mathrm{MsgMLP}(\cdot)$分别表示对环境观测信息$o_n(t)$及对接收消息$m_{n^-}(t-1)$进行处理的全连接层，$\mathrm{Lookup}(\cdot)$表示查表操作[33]。类似的，以下$\mathrm{GRU}(\cdot)$和$\mathrm{MLP}(\cdot)$均表示对应网络结构的运算操作。

经输入网络处理后的嵌入表示$z_n(t)$作为隐藏层网络的输入，此处为了解决环境的部分可观测特性，并且更好地提取环境中的时序变化特征（如信道的时变特性、剩余可传输时间持续减少等），本工作考虑使用循环神经网络来作为神经网络中间隐藏层，具体来说，在本工作仿真实现中采用了两层的GRU作为隐藏层。第一个隐藏层计算过程如下所示：

$$h_n^1(t) = \mathrm{GRU}(z_n(t), h_n^1(t-1)) \tag{4.17}$$

其中$h_n^1(t-1)$表示该隐藏层上一时刻的隐变量。随后第二层GRU网络输出$h_n^2(t)$作为输出层MLP的输入，分别计算出Q值（维度为动作空间维度，即$|\mathcal{S}||\mathcal{A}_P|$）和交互信息（维度为交互消息位数，即$K$），输出层MLP由两层神经元数为128的全连接层组成，且输出维度为$|\mathcal{S}||\mathcal{A}_P| + K$，具体运算如下所示：

$$(Q_n(t), m_n(t)) = \mathrm{MLP}_{[128,128,(|\mathcal{S}||\mathcal{A}_P|+K)]}(h_n^2(t)) \tag{4.18}$$

4.2.4 参数共享

本章节所提出的算法遵循中心式训练-分布式执行的范式，为了充分利用中心式学习的便利，不同智能体间可共享参数，也就是说在训练过程中只需要学习一个共享的网络，所有的智能体都基于这个共享的网络进行决策。参数共享最直接的好处在于其极大地减少了必须学习的参数量，从而加快了训练速度，并且由于本章节中研究的是协作问题，且所有智能体本质上是同构的，因此单个智能体学习到的有效行为模式也可直接推广至所有智能体，有助于加速训练进程。

深度神经网络的强大表征能力可以促进共同策略的学习，同时也允许智能体行为策略产生区分。虽然所有智能体的网络参数一样，但是由于不同智能体对环境的观测不同，即接收的输入不同，其能够相应演化出不同的隐藏状态，因此智能体依然可以表现出不同的行为模式。此外，由于每个智能体还将各自的身份标识（如序号索引n）作为输入，也能促使不同智能体的行为产生区分。

在分布式执行阶段，基于中心式训练阶段得到的共享网络，每个智能体复制得到相应网络副本，维护各自网络的中间隐藏状态，并进行动作决策。在执行阶段只通过广播交互信息来实现与其它智能体的协作。

4.2.5 训练算法

训练过程中，智能体$n \in \mathcal{N}$根据当前环境观测o_n及此前接收到的消息m_{n^-}，每执行一步决策，决定当前动作a_n及发送消息m_n，环境将根据式（4.6）反馈全局奖励r，然后转移到下一时刻状态，智能体n随后获取到新的环境观测o'_n，同时还会接收到其余智能体广播的交互信息m'_{n^-}。智能体每完成一次转移，就将相应的状态转移元组$(o_n, m_{n^-}, a_n, r, o'_n, m'_{n^-})$存储进经验缓存。由于本章节中同样使用了循环神经网络作为隐藏层，因此训练时要求样本为序列存储。此外，为避免多智能体并发训练造成的非平稳特性，以及多智能体策略随着训练进行发生改变导致当前策略与过往样本差距过大，本章节中不使用经典DQN的经验回放技术，当采集了若干个完整转移轨迹后，一并进行批量梯度更新，也就是说单个转移样本只训练一次。为保持算法精简，此处没有采用在上一章节中引入的CERT技术，以展现引入通信机制的效果。

动作选择模块采用经典的Double-DQN方式进行更新。定义TD-error为

$$\Delta Q_n(t) = y_t - Q^{\theta_n}(o_n(t), m_{n^-}(t-1), a_n(t-1), n, a) \tag{4.19}$$

其中

$$y_t = r_t + \gamma Q^{\theta_n^-}(o_n(t+1), m_{n^-}(t), a_n(t), n, \arg\max_{a'} Q^{\theta_n}(o_n(t+1), m_{n^-}(t), a_n(t), n, a')) \quad (4.20)$$

表示Q值更新目标，θ_n^-表示动作选择模块的静态目标网络参数。令α为学习率（更新步长），则动作选择模块的参数更新方式为：

$$\theta \leftarrow \theta + \alpha \frac{\partial}{\partial \theta} (\Delta Q_n(t))^2 \quad (4.21)$$

为了进一步平滑损失函数，此处可采用Huber损失函数对TD-error进行平滑处理[68]：

$$L_\delta(\Delta Q_n(t)) = \begin{cases} \frac{1}{2}(\Delta Q_n(t))^2, & \text{if } |\Delta Q_n(t)| \le \delta \\ \delta|\Delta Q_n(t)| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \quad (4.22)$$

相应参数更新方式为：

$$\theta \leftarrow \theta + \alpha \frac{\partial}{\partial \theta} L_\delta(\Delta Q_n(t)) \quad (4.23)$$

此处δ为一可调超参，反应了损失函数对于异常值的敏感性，δ越接近0，该损失函数对于异常值越不敏感，通常可将其设置为1。

对于智能体n的消息产生模块，由于其当前时刻输出交互消息将作为下一时刻其它智能体的输入，因此，从端到端训练的角度可定义其损失函数为其它所有智能体在未来的Q值评估误差累积。定义$\hat{m}_n(t)$为智能体n的消息产生模块在t时刻经DRU处理后的实际输出，即$\hat{m}_n(t) = \mathrm{DRU}(m_n(t))$，则由链式法则可得消息产生模块的梯度更新规则如下：

$$\theta \leftarrow \theta + \alpha \mu_n(t) \frac{\partial \mathrm{DRU}(m_n(t))}{\partial m_n(t)} \frac{\partial m_n(t)}{\partial \theta} \quad (4.24)$$

其中$\mu_n(t)$表示其它智能体未来Q值评估误差相对于接收到的智能体n发送的广播交互信息$\hat{m}_n(t)$的梯度累积：

$$\mu_n(t) = \sum_{n' \in \mathcal{N}, n' \neq n} \frac{\partial}{\partial \hat{m}_n(t)} (\Delta Q_{n'}(t+1))^2 \quad (4.25)$$

在训练过程中将上述动作选择模块和消息产生模块的梯度分别进行累积后，即可对神经网络参数θ进行更新。由于在训练过程中采用了静态目标网络技术，即额外使用一个网络θ_n^-来辅助评估Q值，因此该目标网络需要以较低的频率来更新参数，通常每隔N_U步，就将评估网络参数复制到目标网络中：

$$\theta_n^- \Longleftarrow \theta_n \quad (4.26)$$

在智能体训练和执行阶段，算法唯二的区别在于：1）是否采用ϵ-greedy策略选择动作，在执行阶段无需进行探索以获取多样的训练样本，因此直接选取Q值最大的动作即可；

以及2）DRU单元对消息产生模块输出的处理上，在执行阶段为符合实际通信系统，需要对消息进行离散化。此处通过逐元素根据$m_n(t)$取值是否大于0来进行离散化，实际输出为：

$$\hat{m}_n(t) = \mathbb{1}(m_n(t) > 0) \tag{4.27}$$

此处，$\mathbb{1}(\cdot)$为逐元素操作，因此执行阶段消息产生模块的最终输出为K比特的0-1矢量。

综上所述，本章节所提出的算法训练流程总结于下页算法 4.1中。具体来说，算法4.1的第4至9行描述了智能体与环境的互动过程。在每个时间步t中，每个智能体n根据式(4.14)表示的ϵ-greedy策略选择动作$a_n(t)$，以平衡探索和利用，并获得环境反馈奖励。第10行到第13行描述了状态转移及将转移样本存储到缓存中的过程。第15行至第24行描述了智能体动作选择模块及消息产生模块参数更新的过程。最后，第25行表示静态目标网络以较低的频率更新，以使评估网络的更新目标保持稳定。 注意，当采用参数共享机制时，所有智能体共享网络参数θ和θ^-，且只需要训练一个网络即可。

4.3 仿真结果

本章节将给出仿真结果以验证该结合了智能体间通信机制的MARL算法应用于车联网分布式频谱接入问题的性能表现。首先介绍基本的仿真设置，随后验证本章节所提出算法的有效性。为探究本章节所引入的通信机制对算法性能的影响，本章节将研究进行信息交互时智能体所能发送的消息比特数不同时，算法性能的变化情况。在本章节设计的消息产生模块中引入了DRU模块，其中在训练阶段主动引入了噪声，此处也将研究该噪声对算法训练结果的影响。最后，本章节还将研究该算法对于实际环境中可能出现的交互信息传输出错情况的稳健性。

4.3.1 仿真设置

此处仿真设置和章节3.3基本保持一致，其中道路拓扑模型和信道链路模型完全保持一致，分别如图3.1和表3.3所示。仿真参数设置也基本保持一致，只是本章节中考虑V2V-VUE以更高的频率发送安全相关消息，因此时延约束更小，此处设置$T_{\max} = 20$ms，相应的数据包大小设置为$\{1, 2, \ldots, 6\} \times 210$字节。具体参数设置见表4.1。

训练过程中采用的超参数如表 4.2所示。折扣率γ通常可设置为1减去回合数据长度的倒数（即$1 - \frac{1}{20} = 0.95$）。在训练过程中，探索率ϵ逐渐降低以平衡探索和利用。此处同样采取了第3.2.2节中提出的滞后学习机制，采用滞后学习率β来使智能体在训练初期保持乐

算法 4.1 多智能体通信决策联合学习算法

输入： 学习率α，探索率ϵ，折扣系数γ，目标网络更新频率N_{U}。

为每个智能体$n \in \mathcal{N}$随机初始化网络参数θ_n，并且将其复制为目标网络参数θ_n^-；
for 训练中的每一幕轨迹e **do**
  for 每一时间步t **do**
    for 每一个V2V-VUE智能体n **do**
      获取对环境的观测$o_n(t)$并接收其余智能体交互消息$m_{n^-}(t-1)$；
      根据ϵ-greedy策略选择动作$a_n(t)$；
      产生消息$m_n(t)$；
    end for
    获取全局奖励r_t；
    for 每一个V2V-VUE智能体n **do**
      获取新的环境观测$o_n(t+1)$以及交互信息$m_{n^-}(t)$；
      将状态转移元组$(o_n(t), m_{n^-}(t-1), a_n(t), r_t, o_n(t+1), m_{n^-}(t))$存储进缓存；
    end for
  end for
  for 每一个V2V-VUE智能体n **do**
    置零梯度$\nabla\theta$；
    for 每一时间步t **do**
      对转移样本$e=(o,m,a,r,o',m')$计算目标值：
      $$y = r + \gamma Q^{\theta_n^-}(o', m', \underset{a'}{\operatorname{argmax}} Q^{\theta_n}(o', m', a'))；$$
      计算TD-error: $\Delta Q_n(t) = y_e - Q^{\theta_n}(o, m, a)$；
      为动作选择模块累计梯度：$\nabla\theta \Longleftarrow \nabla\theta + \frac{\partial}{\partial\theta} L_\delta(\Delta Q_n(t))$；
      为消息产生模块更新梯度计算链：
      $$\mu_n(t) = \sum_{n'\in\mathcal{N}, n'\neq n} \frac{\partial}{\partial \hat{m}_n(t)} L_\delta(\Delta Q_{n'}(t+1))；$$
      为消息产生模块累计梯度：
      $$\nabla\theta \Longleftarrow \nabla\theta + \mu_n(t) \frac{\partial \mathrm{DRU}(m_n(t))}{\partial m_n(t)} \frac{\partial m_n(t)}{\partial \theta}；$$
    end for
    更新网络参数：$\theta_n \Leftarrow \theta_n + \alpha\nabla\theta$；
    每N_{U}步更新目标网络参数：$\theta_n^- \Longleftarrow \theta_n$；
  end for
end for

观的策略。由于在训练后期，当每个智能体都学习到了较好的策略后，评估的准确性将变得更加关键，因此，β将随着训练进行逐渐增加以平衡正负样本之间的更新。由于本章节中所提出的算法没有采用经验回放机制，因此没有CERT相关参数。取而代之的是在训练过程中，先使智能体完整地经历N_{B}个回合，再一并进行批量梯度更新。由于本章节中奖励函数（见章节4.1.2）经过了重新设计，因此相应的奖励函数权重也需要进行调整，具体超参数设置见表4.2所示。

在分布式执行阶段，每个智能体感知对环境的局部观测结果，并根据训练得到的统一模型选择具有最大Q值的动作。在训练阶段V2V-VUE的传输数据大小固定为$L = 6 \times 210$字

表 4.1 仿真参数设置

参数	值
载波频率	2 GHz
子信道带宽	1 MHz
基站天线高度	25 m
基站天线增益	8 dBi
基站接收机噪声系数	5 dB
车辆天线高度	1.5 m
车辆天线增益	3 dBi
车辆接收机噪声系数	9 dB
车辆速度	$[10, 15]$ m/s
子信道个数 S	4
V2I发射机功率 $P_m^{\mathrm{I}}, m \in \mathcal{M}$	23 dBm
V2V发射机功率$P_n^{\mathrm{V}}, n \in \mathcal{N}$	$\{-100, 5, 15, 23\}$dBm
噪声功率 σ^2	-114 dBm
V2I最小吞吐量$R_{\min}^{\mathrm{I}}$	10 Mbps
V2V时延约束$T_{\max}$	20 ms
V2V数据包大小L	$\{1, 2, \ldots, 6\} \times 210$ bytes

表 4.2 训练超参数设置

参数	值
学习率α	0.0001
折扣率γ	0.95
探索率ϵ	$1.0 \rightarrow 0.1$
滞后学习率β	$0.2 \rightarrow 0.8$
总探索回合数	3500
总训练回合数	5000
批大小N_{B}	8
目标网络更新频率N_{U}	2
奖励权重$\{\lambda_i, i \in \{1, 2, 3, 4\}\}$	$\{0.10, 0.12, 1.0, 1.0\}$
激励常数c	5
Huber损失函数参数δ	1
交互控制信道带宽K	2

节，在执行阶段可以变化，以验证算法泛化性。此外，为了获得更多样的训练样本，训练过程中会定期重新初始化车辆的位置。

本章节中由于输入包含多种不同类型的信息，所以神经网络的输入结构需要进行针对性调整，此前在章节4.2.3中已经做出了相应描述，具体的网络结构及配置如表4.3所示。

4.3.2 性能验证

本章节训练过程中同样采用Random和Round-Robin策略作为基准，来计算式（4.6）中

表 4.3　神经网络结构

输入层	两个嵌入层（分别用于智能体ID和之前的动作编号），两个全连接层（分别用于环境观测和接收到的消息），输出维度均为128
隐藏层	两层GRU层，隐藏层维度为128
输出层	两层全连接层，维度分别为128，18
激活函数	修正线性单元（ReLU）

定义的奖励函数。具体来说，在Round-Robin方法中，传输功率固定为最大值，子信道选择进行轮转；而在Random方法中，子信道和传输功率都是随机选择。

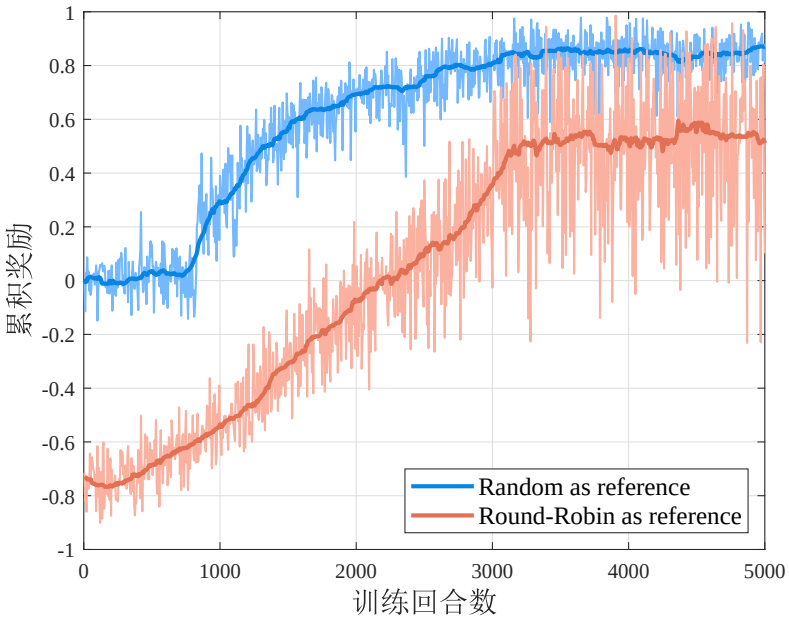


图 4.3　训练过程中的累积奖励变化曲线

图 4.3展示了归一化后的训练过程中智能体累积奖励的变化曲线。从图 4.3中可以看到，基于Random和Round-Robin作为参考基准的算法累积奖励曲线随着训练进行基本都能够收敛。此处可以观察到使用Random作为基准策略的奖励函数略高，这是因为Random方法相比Round-Robin方法作为基准计算式（4.6）中的奖励时，其基准值会更低一些，因此智能体计算实际获得的奖励时相应更高。

图 4.4展示了以Round-Robin为基准时，本章节所提出算法训练过程中的V2V-VUE包交付率的变化曲线。可以看到随着训练进行，V2V-VUE包交付率有了明显的增长。另外可以注意到V2V-VUE包交付率曲线的变化趋势与图 4.3比较一致，这是因为在设置奖励函数的权重时，相应地为V2V-VUE赋予了更高的权重，因此V2V-VUE相关指标对奖励函数影响较大。

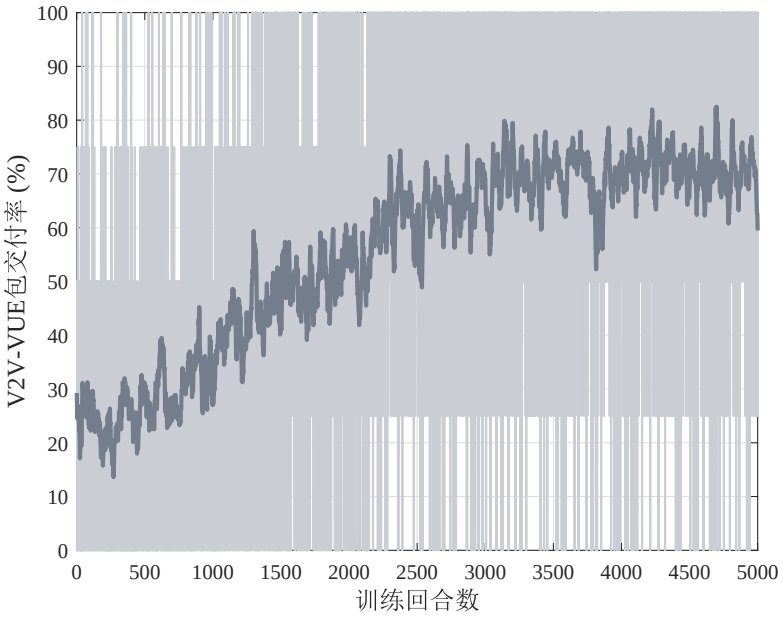


图 4.4　训练过程中的V2V-VUE包交付率变化曲线

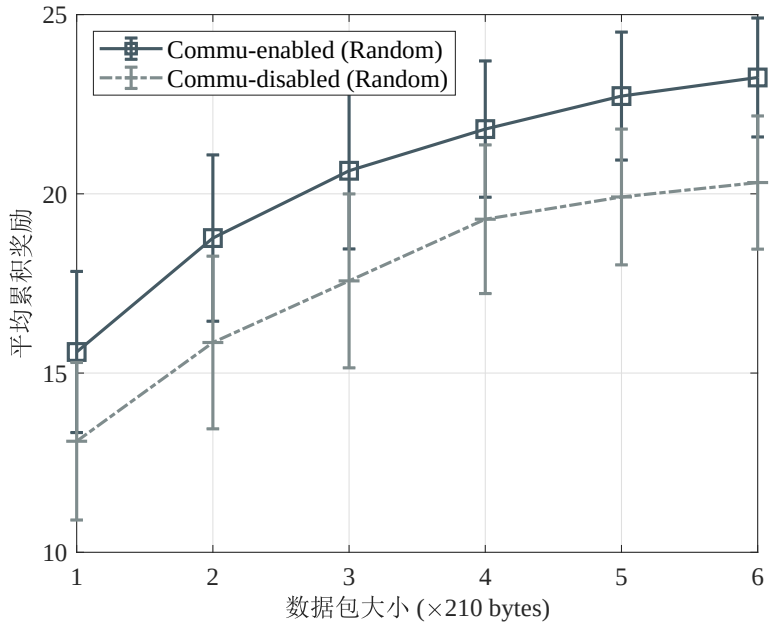


图 4.5　累积奖励对比（使用Random作为基准）

接下来，本章节所提出的算法将与两类策略进行对比，一种是完全启发式的，即训练过程中作为基准策略的Random和Round-Robin策略；此外还将与同样基于多智能体强化学习设计的分布式频谱接入算法3.1进行对比。由于在该算法中各智能体在执行阶段是完全独立工作的，没有通信机制辅助协作，因此在接下来的仿真结果中分别用*Commun-enabled*和*Commun-disabled*，来指代本章提出的算法4.1和第3章中所提出的算法3.1。

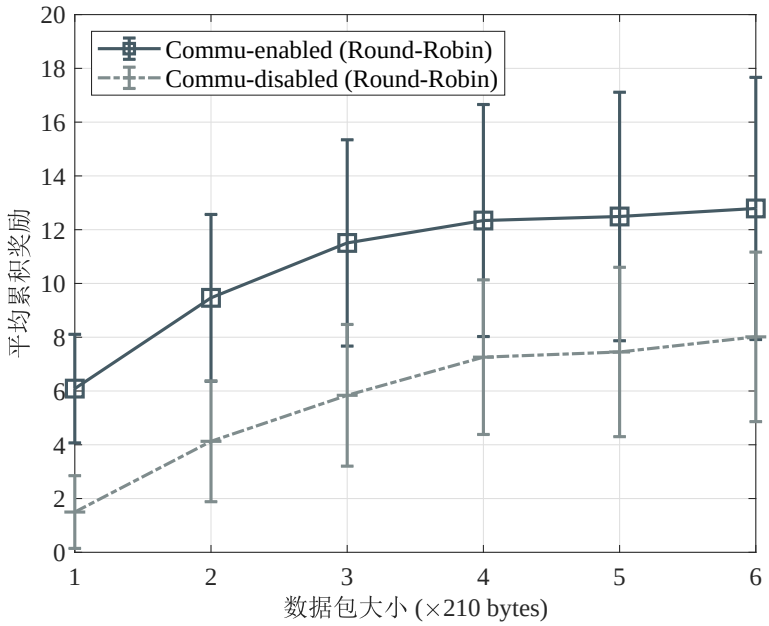


图 4.6　累积奖励对比（使用Round-Robin作为基准）

首先将本章引入通信机制算法的性能同上一章没有通信机制的算法进行对比，图4.5和图 4.6分别展示了采用Random和Round-Robin作为基准进行训练时，在测试阶段所获得的累积奖励。从图 4.5和图 4.6中可以看出本章提出的引入通信机制的算法相较于上一章所提出的智能体间完全独立工作的算法能够获得更多奖励。此外，从图 4.5和图 4.6中可以注意到智能体获得的平均累积奖励随着V2V-VUE发送数据包的大小增大而同步增加，这是因为当数据包增大时，基准方案（Random或Round-Robin）难以在时延约束前完成发送，与所提出的算法性能表现差距相应增大。

接下来具体对所关注的V2I-VUE和V2V-VUE相关指标进行分析。图 4.7展示了V2I-VUE总吞吐量随着V2V-VUE传输数据量大小增加的变化趋势。当传输数据量增加时，V2V-VUE需要花费更长的时间才能完成传输，因此为V2I-VUE带来的干扰持续时间也随着增加，所以图 4.7中所有方案的性能都相应有所下降。从图 4.7中可以看出，本章节和上一章节中所提出的同样基于MARL的算法性能均优于相应的Random和Round-Robin基准策略。通过对比Commun-enabled曲线和Commun-disabled曲线可以看出，本章节所提出的引入智能体间通信机制的算法相较于上一章所提出的智能体间完全独立工作的算法版本具有一定性能优势，能够获得更高的V2I-VUE总吞吐量。

图 4.8展示了V2V-VUE的数据包交付率随着传输数据包大小增加的变化关系，随着传输数据包的增加，所有方案的包成功交付率指标同样会有一定的下降。从图 4.8中可以观察到本章节所提出算法相对于作为基准的Random方案和Round-Robin方案的显著优势。虽

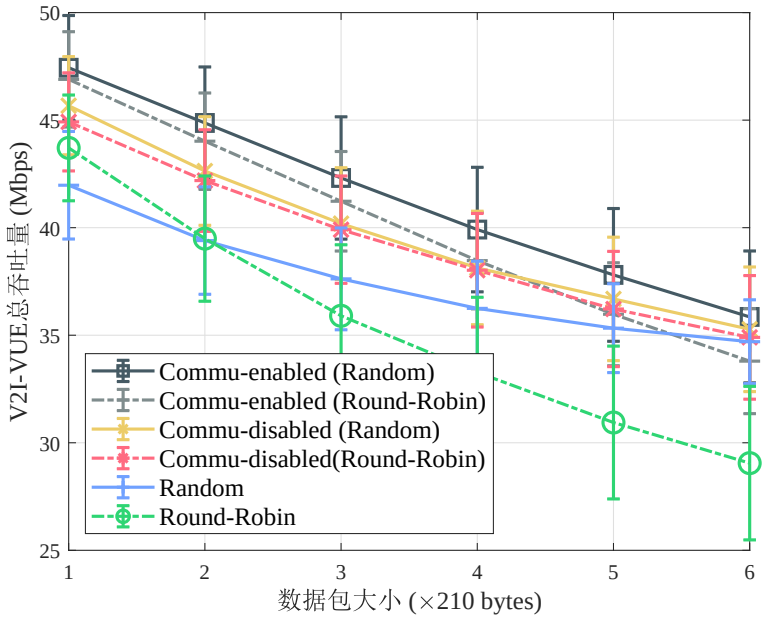


图 4.7　V2I-VUE的总吞吐量

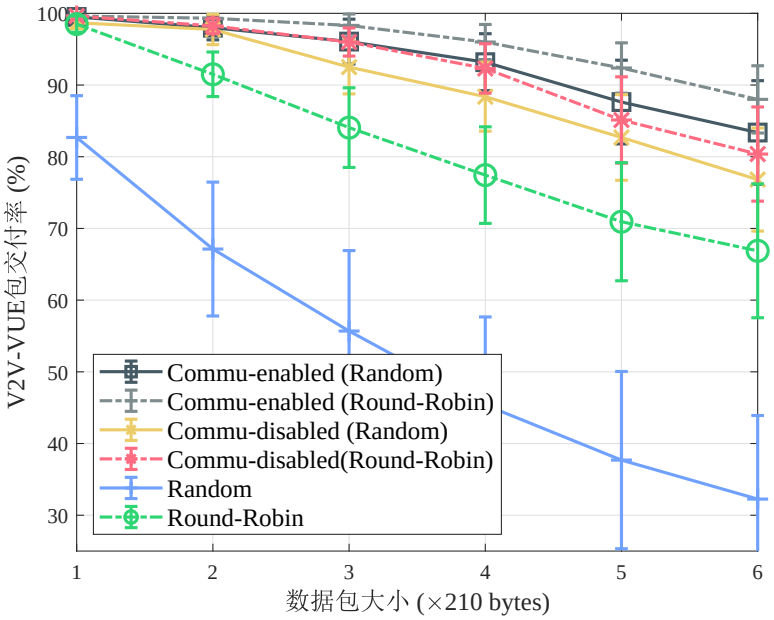


图 4.8　V2V-VUE的包交付率

然此前第3.3章中已经展现出了基于MARL的分布式频谱接入算法具备比较良好的性能，但从图 4.8中可以观察到，本章节所提出的引入通信交互机制的算法能够带来进一步的性能增益。另外需要澄清的是，图 4.8中呈现的Commu-disabled曲线基本是完全由第3章中所提出的算法得到，只是针对章节4.1.2中修改设计的奖励函数进行重新训练。之所以此处性能相对于图 3.8中呈现的结果有一定下降，是因为在本章节中的仿真设置中，考虑V2V-VUE以更高的频率发送消息，即智能体需要在更短的时间内完成数据包发送，虽然

数据量相应进行了减小，但是智能体能够相互协调的时间余地同样减小了，所以会有一定的性能损失。

4.3.3　通信带宽对算法性能影响

在本章节的系统设置中，假设智能体间用于信息交互的广播信道带宽为K比特，即每个智能体可广播K维的0-1矢量信息用于辅助协作。直观来看，可交互的信息越多，智能体间协作效果也会相对更好，因此本小节将验证交互信道带宽对算法性能的影响，以下分别比较交互消息比特数为1、2、4时，测试阶段智能体平均累积奖励、V2I-VUE的总吞吐量及V2V-VUE的包交付率三个指标。值得注意的时，当消息比特数为0时，即智能体间没有交互机制，本章节所提出的算法基本简化为上一章节所提出的完全独立式算法。

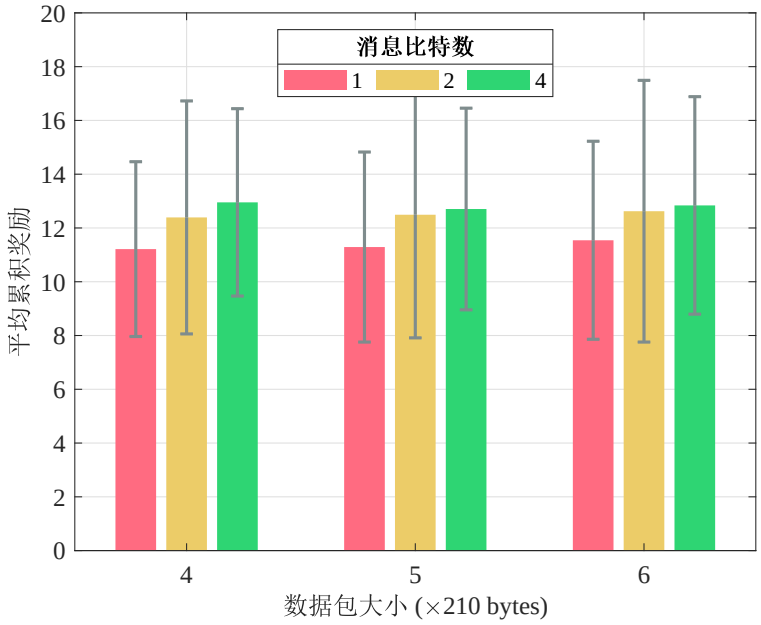


图 4.9　通信比特数对累积奖励的影响

图 4.9展示了通信比特数对智能体累积奖励的影响，可以看出随着智能体间交互比特的增加，累积奖励基本呈现增长趋势，这也符合直观上的认识。此外，从图 4.9中还可以看出，当消息比特数从1增加到2时，累积奖励指标提升的幅度明显大于当消息比特数从2增加到4时，因此可以判断出增大交互信息量对于本章节提出的引入通信机制的算法性能提升存在边际效应。

图 4.10和图 4.11分别展示了交互信息比特数对V2I-VUE总吞吐量和V2V-VUE包交付率两项指标的影响。由于V2I-VUE的总吞吐量指标和V2V-VUE的包交付率指标间存在权衡关

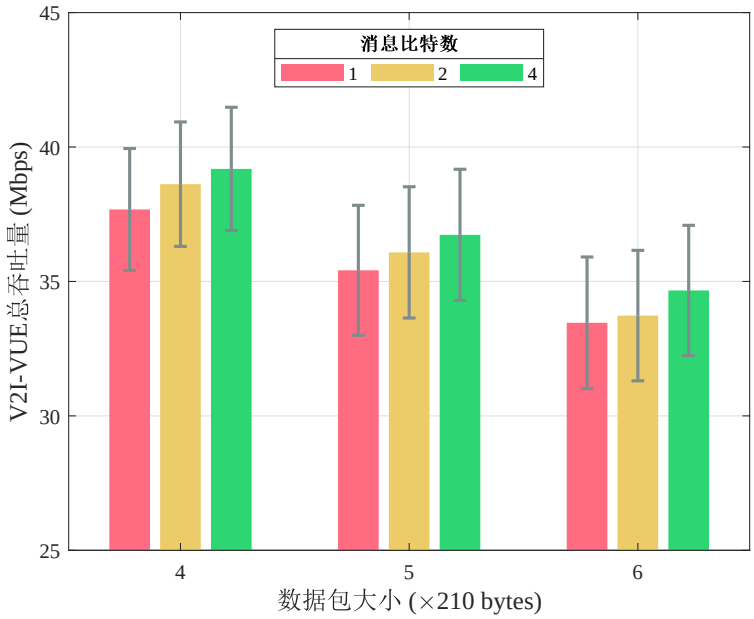


图 4.10　通信比特数对V2I-VUE总吞吐量的影响

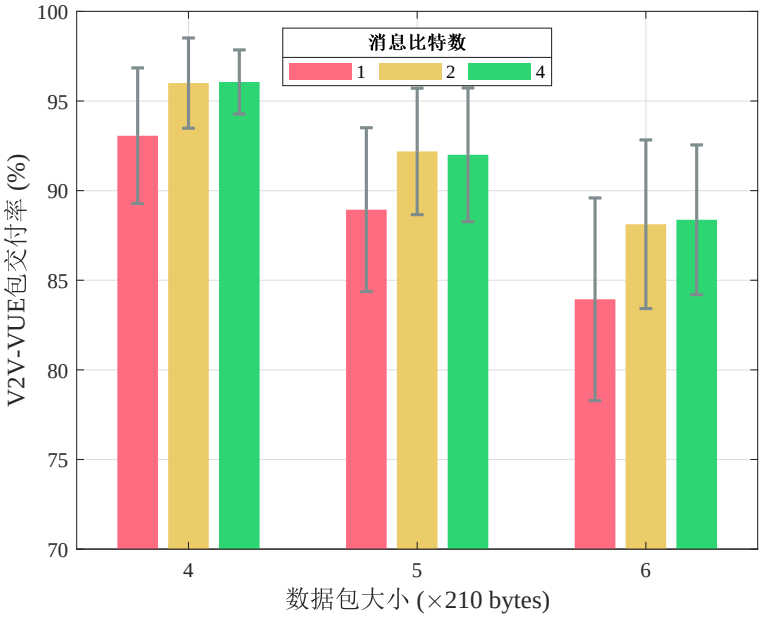


图 4.11　通信比特数对V2V-VUE包交付率的影响

系，因此从图 4.10和图 4.11中可以看到，可能出现随着通信比特数增加，V2I-VUE的总吞吐量上升，而V2V-VUE的包交付率指标略微下降的情况。但从图 4.10和图 4.11中呈现的总体趋势来看，还是可以得到和此前图 4.9类似的结论：消息比特数的增加有助于提升算法性能，但是存在边际效应。

4.3.4　引入噪声对算法性能影响

为了实现消息产生模块的端到端训练，在章节4.2.2中引入了DRU机制。DRU在训练阶段将消息产生模块的连续输出正则化，而在测试阶段将消息产生模块输出离散化。为了减少从连续实值映射到0-1比特时产生的离散化误差，DRU在训练阶段为其输入加上一定的高斯白噪声，以限制消息产生模块连续输出可表示的信息量。经过实验发现，高斯白噪声的大小对于训练效果影响重大。

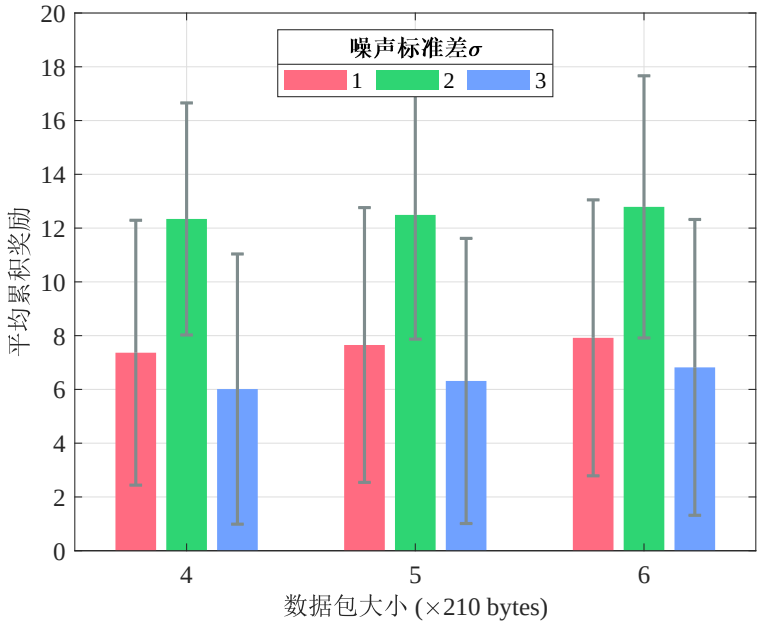


图 4.12　引入噪声大小对累积奖励的影响

图 4.12、4.13和4.14分别表示在训练阶段DRU引入噪声标准差σ为1、2和3时，测试阶段该算法累积奖励、V2I-VUE总吞吐量和V2V-VUE包交付率三项指标的变化情况。从图4.12中可以看出，训练阶段DRU噪声不宜太大也不宜太小，$\sigma = 2$可能是比较合适的设置。如果在训练过程中引入的噪声太小，则消息产生模块的连续实值输出可表征的信息量较大，当在测试阶段离散化为0-1比特时，会产生较大的离散误差；而当噪声较大时，则会淹没智能体原本想要发送的交互信息，同样不利于学习。从图 4.14中可以观测到类似的结论。而由于V2I-VUE总吞吐量指标和V2V-VUE包交付率指标存在权衡关系，因此图4.13中会出现当$\sigma = 2$时，V2I-VUE总吞吐量指标相对较差的情况。

4.3.5　通信差错对算法性能影响

在此前的仿真设置中，均假设智能体广播的交互信息将无差错地进行传输。由于每个

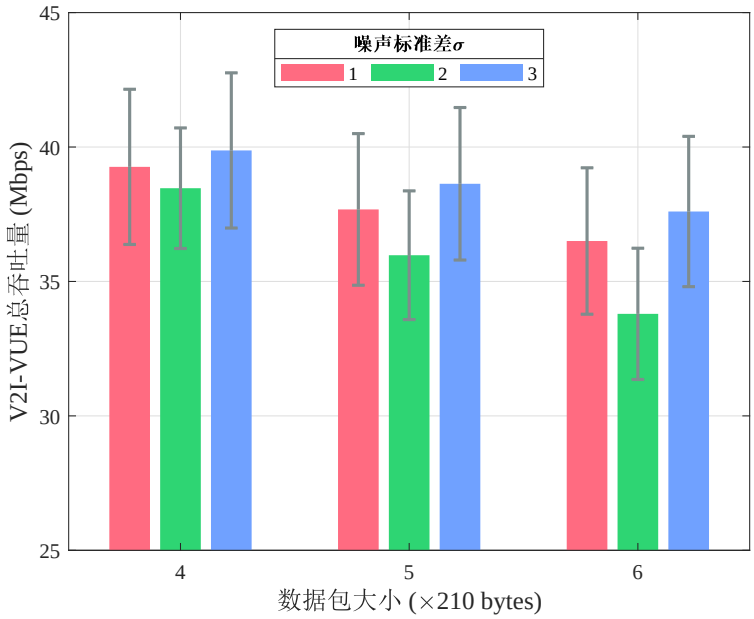


图 4.13　引入噪声大小对V2I-VUE总吞吐量的影响

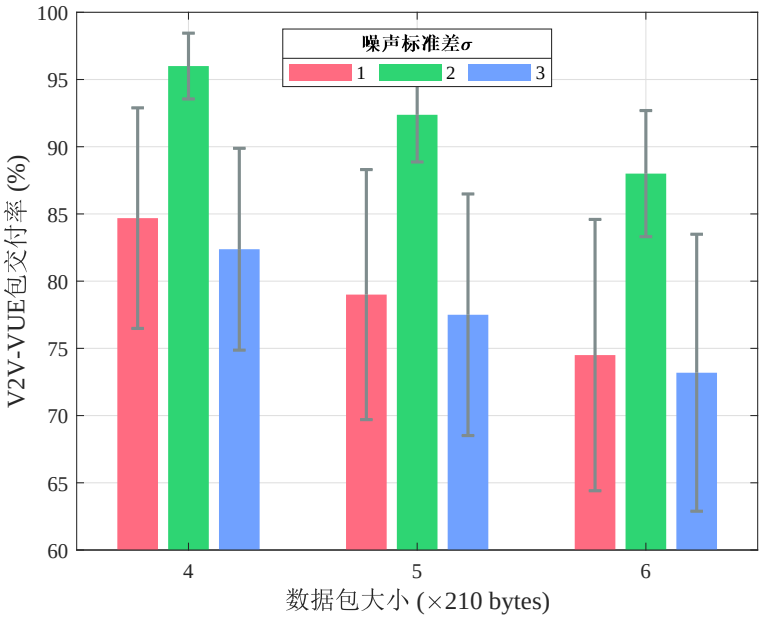


图 4.14　引入噪声大小对V2V-VUE包交付率的影响

智能体发送的交互信息比特数很少，因此若采用较低阶数的调制方式的话在实际系统中也基本能够满足该假设。为全面考量算法性能，本小节将研究智能体交互信息传输出错时，算法性能的变化情况。

图 4.15、4.16和4.17分别表示在测试阶段智能体广播交互信息以不同出错概率进行传输时，累积奖励、V2I-VUE总吞吐量和V2V-VUE包交付率三项指标的变化情况。由于在

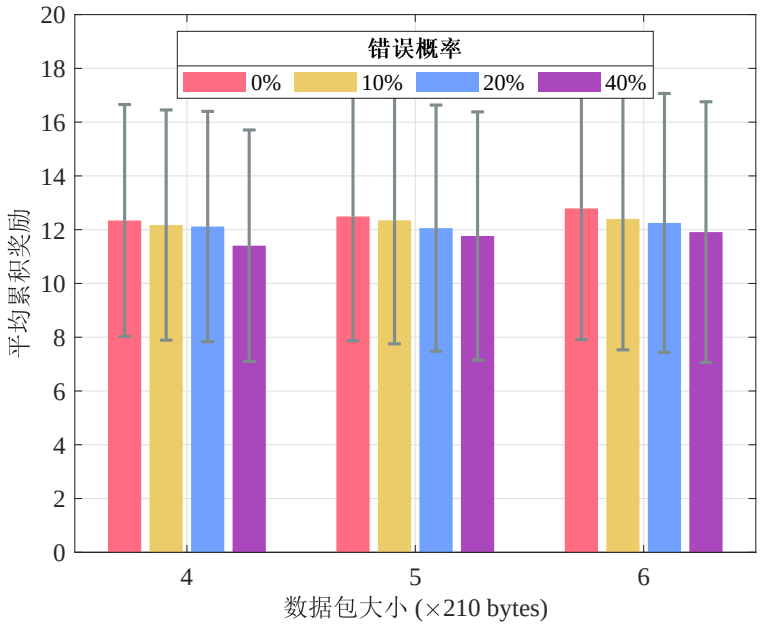


图 4.15　通信差错概率对累积奖励的影响

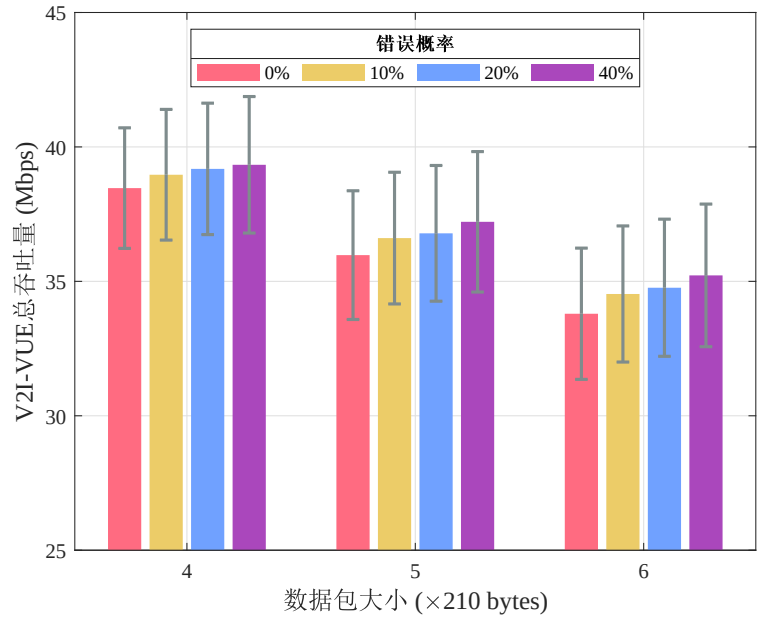


图 4.16　通信差错概率对V2I-VUE总吞吐量的影响

测试阶段，智能体发送的交互信息均经过DRU单元离散处理为0-1比特，因此这里的传输出错指智能体接收到的比特相对于原本发送信息的比特以一定概率发生翻转。从图4.15和图 4.17中可以观察到，随着传输差错概率的增大，该算法的性能均发生了一定程度的下降，说明本章节所提出的引入通信机制的算法性能会受到交互信息信道质量的影响。此外，由于V2I-VUE总吞吐量指标和V2V-VUE包交付率指标之间存在权衡关系，从

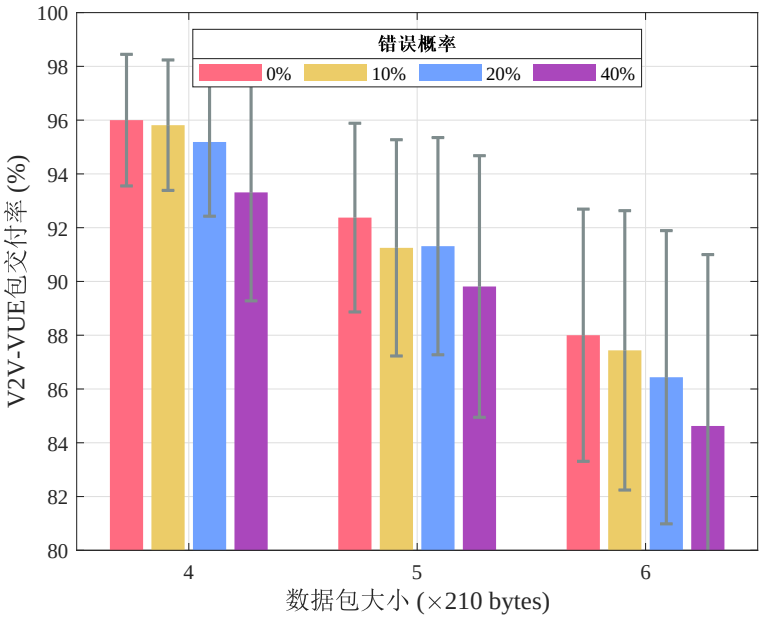


图 4.17　通信差错概率对V2V-VUE包交付率的影响

图 4.16和图 4.17中可以观察到，随着错误概率增加，虽然V2V-VUE包交付率下降了，但此时V2I-VUE总吞吐量却相应有所上升。不过也可以看出，该算法对于交互信息出错有一定的稳健性，考虑到实际场景中使用较低调制阶数时通信系统的错误概率本身就较低，因此此处算法的性能下降完全在可接受范围内。

4.4　本章小结

本章节考虑车联网中V2I-VUE和V2V-VUE共存的场景，以最大化V2I-VUE总吞吐量，同时满足V2V-VUE时延可靠性要求为优化目标设计分布式频谱接入机制。本章节基于多智能体强化学习，提出了一种引入智能体间通信机制的分布式频谱接入控制算法。通过引入通信机制，智能体间能够显式地进行交互，来实现更好的任务协作。在该算法中，智能体由动作选择模块和消息产生模块组成：动作选择模块完成频谱接入动作的选择，而消息产生模块则负责生成交互信息。动作选择模块和消息产生模块均由神经网络实现，且共享网络参数。通过引入DRU结构对消息产生模块的输出在训练阶段正则化，而在测试阶段离散化，可以使得智能体的神经网络参数可以执行端到端训练，且减小在测试阶段的性能损失。在训练阶段，所有智能体共享网络参数，以减小训练开销。最终，仿真结果验证了本章节所提出算法引入的通信机制的有效性，相较于在上一章节提出的完全独立工作的算法版本，有一定的性能提升。

5 总结与展望

本章节将对论文整体内容进行总结，并对潜在的未来研究方向进行展望。

5.1 工作总结

本学位论文主要研究车联网中的分布式频谱接入算法设计，考虑城市道路中V2I用户和V2V用户共存场景下，以最大化V2I用户总吞吐量同时满足V2V用户时延需求为优化目标，基于多智能体强化学习设计分布式的频谱接入算法。

将深度强化学习，乃至多智能体强化学习应用于通信领域中的资源分配问题是当下的一个研究热点，然而如章节1.2所述，现有工作存在着诸多的不足，例如没有考虑多智能体同时训练产生的非平稳性、忽视了由于车辆移动性导致的车联网环境动态特性将会发生变化等实际问题。因此本文在第三章中基于多智能体强化学习设计完全独立工作的分布式频谱接入算法时，引入了一系列学习技术。首先，考虑到车联网信道的时变特性，以及环境的部分可观测性，在网络结构设计中采用RNN作为隐藏层，并且结合Dueling-DQN结构，实现更好的值函数估计。此外，还结合了Double-DQN技术来解决值过估计问题。以上各技术组成了第三章中所提出的D3RQN算法。为了应对多智能体并发学习所引起的非平稳性，该章节在设计算法的训练过程中引入了滞后学习机制和并发经验回放机制，实现更稳定的训练过程。为了解决训练过程中环境动态特性会发生变化的问题，该算法引入了一种近似遗憾奖励机制，通过引入静态基准策略来辅助实现更准确的训练效果评估。在设计算法的观测空间时，考虑到实际环境中的限制，本文排除了获取CSI的必要，最终仿真结果表明，即使不使用CSI反馈，仅使用自身对环境的局部观测结果，该算法依然能得到可观的性能表现，并且表现出了良好的稳健性和可扩展性。

将通信机制引入多智能体强化学习则是当下较为新颖的研究领域，通过引入通信机制，使智能体在决策过程中显式地进行交互，能够有效地使智能体更好地进行协作。尽管强化学习社区近年来已经涌现出了诸多具备影响力的工作，但通信领域目前仍鲜有涉及。在第三章提出的智能体间完全独立工作的分布式频谱接入算法之外，本文在第四章创新性

地将结合了通信机制的多智能体强化学习应用于车联网分布式频谱接入算法设计中。在该章节的算法设计中，智能体不仅需要决策频谱接入动作，还需要生成交互信息。智能体由动作选择模块和消息产生模块组成，均由神经网络实现，且两个模块共享参数。为了最大化利用中心式的训练模式，训练阶段设置智能体发送的交互信息为连续值，因此可使用损失函数梯度回传对消息产生模块的参数进行端到端更新。为了符合实际通信系统，在分布式执行阶段，智能体交互的信息为离散比特，因此在算法设计中引入了离散/正则处理单元对消息产生模块的输出进行处理。在实际训练过程中，为节省训练开销，所有智能体共享同一套网络参数。最终，仿真结果表明，引入通信机制能够帮助智能体实现更好的协作效果。

总的来说，本学位论文将多智能体强化学习应用于车联网分布式频谱接入算法的设计之中，展示了多智能体强化学习用于解决多用户参与、分布式序列决策问题的有效性。

5.2 未来展望

在进行本学位论文研究内容的过程中，结合当下的研究热点，作者认为有以下潜在的研究方向供未来改进和完善：

1) 泛化性是如今众多强化学习算法面临的巨大难题[69]，而在真实的车联网环境中，车流量模型、车辆移动模式、信道模型乃至道路拓扑，都将变得更加复杂，要想将基于多智能体强化学习设计的频谱接入算法应用于真实场景，无疑对算法的泛化性能有着极高的要求。所以，如何提升泛化性将是未来一个重要的研究方向。

2) 在真实世界的道路场景中，车辆密度通常是较高的，因此基于多智能体强化学习设计算法时，需要处理智能体数目较大时难以训练的问题，对算法训练的效率，以及可扩展性要求较高。文章[70]中提出的平均场强化学习，通过采用平均建模的思想，能够适用于智能体数目较大的情况，可能是一个有效的解决途径。

3) 本论文中，均基于经典的DQN算法基础之上进行改进，由于DQN属于基于值函数的强化学习算法，因此动作空间需要离散化处理，而在实际场景中，类似传输功率等控制动作的取值可为连续值，离散化处理可能丢失一定的控制精度，因此未来可采用更细致的离散间隔，乃至直接采用基于策略的强化学习算法，如DDPG、PPO等，进行算法设计。

4) 本文第四章中，将结合了通信机制的多智能体强化学习算法应用于车联网频谱算法设计之中。然而，脱离具体的应用场景，作者认为结合通信机制的多智能体强化学习算法在通信领域有着广泛的潜在应用场景。可以认为用户间进行通信的目的是为了完成特定任务[40]，基于此思想，使用多智能体强化学习技术，可以实现行为策略和通信协议的联合学习，其与传统通信系统设计中以减少传输差错概率为目标是不同维度的设计思路。更具体来看，这也为无线通信系统中的联合编码设计提供了一种新的思路。

总而言之，多智能体强化学习本身就是一个非常活跃的研究领域，而由于车联网这样的无线通信系统天然具备多用户特性，因此多智能体强化学习应用于无线通信系统是非常契合的，存在很多可供挖掘的方向。

参考文献

[1] 中华人民共和国国家统计局. 国家数据[OL]. https://data.stats.gov.cn/easyquery.htm?cn=C01&zb=A0G0I&sj=2020, 2020.

[2] Mate Boban, Apostolos Kousaridas, Konstantinos Manolakis, Josef Eichinger, Wen Xu. Connected roads of the future: Use cases, requirements, and design considerations for vehicle-to-everything communications[J]. IEEE Vehicular Technology Magazine, 2018. 13(3):110–123.

[3] 前瞻产业研究院. 2021-2026年中国车联网行业市场前瞻与投资战略分析报告[R]. 北京:, 2021.

[4] 交通运输部. 交通运输部关于推动交通运输领域新型基础设施建设的指导意见[OL]. http://www.gov.cn/zhengce/zhengceku/2020-08/06/content_5532842.htm, 2020.

[5] 中国信息通信研究院. 车联网白皮书（C-V2X分册）[R]. 北京:, 2019.

[6] Sohan Gyawali, Shengjie Xu, Yi Qian, Rose Qingyang Hu. Challenges and solutions for cellular based V2X communications[J]. IEEE Communications Surveys & Tutorials, 2021. 23(1):222–255.

[7] Hongli He, Hangguan Shan, Aiping Huang, Long Sun. Resource allocation for video streaming in heterogeneous cognitive vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2016. 65(10):7917–7930.

[8] Chunhua Hong, Hangguan Shan, Meiyan Song, Weihua Zhuang, Zhiyu Xiang, Yingxiao Wu, Xiaoli Yu. A joint design of platoon communication and control based on LTE-V2V[J]. IEEE Transactions on Vehicular Technology, 2020. 69(12):15893–15907.

[9] Xiaoshuai Li, Lin Ma, Rajan Shankaran, Yubin Xu, Mehmet A. Orgun. Joint power control and resource allocation mode selection for safety-related V2X communication[J]. IEEE Transactions on Vehicular Technology, 2019. 68(8):7970–7986.

[10] Le Liang, Hao Ye, Guanding Yu, Geoffrey Ye Li. Deep-learning-based wireless resource allocation with application to vehicular networks[J]. Proceedings of the IEEE, 2020. 108(2):341–356.

[11] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, Joelle Pineau. An introduction to deep reinforcement learning[J]. Foundations and Trends® in Machine Learning, 2018. 11(3-4):219–354.

[12] Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, Dong In Kim. Applications of deep reinforcement learning in communications and networking: A survey[J].

IEEE Communications Surveys & Tutorials, 2019. 21(4):3133–3174.

[13] Xianfu Chen, Zhifeng Zhao, Celimuge Wu, Mehdi Bennis, Hang Liu, Yusheng Ji, Honggang Zhang. Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach[J]. IEEE Journal on Selected Areas in Communications, 2019. 37(10):2377–2392.

[14] Hoon Lee, Tony Q. S. Quek, Sang Hyun Lee. A deep learning approach to universal binary visible light communication transceiver[J]. IEEE Transactions on Wireless Communications, 2020. 19(2):956–969.

[15] Hoon Lee, Sang Hyun Lee, Tony Q. S. Quek. Deep learning for distributed optimization: Applications to wireless resource management[J]. IEEE Journal on Selected Areas in Communications, 2019. 37(10):2251–2266.

[16] Shangxing Wang, Hanpeng Liu, Pedro Henrique Gomes, Bhaskar Krishnamachari. Deep reinforcement learning for dynamic multichannel access in wireless networks[J]. IEEE Transactions on Cognitive Communications and Networking, 2018. 4(2):257–265.

[17] Oshri Naparstek, Kobi Cohen. Deep multi-user reinforcement learning for distributed dynamic spectrum access[J]. IEEE Transactions on Wireless Communications, 2019. 18(1):310–323.

[18] Yiding Yu, Taotao Wang, Soung Chang Liew. Deep-reinforcement learning multiple access for heterogeneous wireless networks[J]. IEEE Journal on Selected Areas in Communications, 2019. 37(6):1277–1290.

[19] Helin Yang, Zehui Xiong, Jun Zhao, Dusit Niyato, Chau Yuen, Ruilong Deng. Deep reinforcement learning based massive access management for ultra-reliable low-latency communications[J]. IEEE Transactions on Wireless Communications, 2021. 20(5):2977–2990.

[20] Hao Ye, Geoffrey Ye Li, Biing-Hwang Fred Juang. Deep reinforcement learning based resource allocation for V2V communications[J]. IEEE Transactions on Vehicular Technology, 2019. 68(4):3163–3173.

[21] Xinran Zhang, Mugen Peng, Shi Yan, Yaohua Sun. Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications[J]. IEEE Internet of Things Journal, 2020. 7(7):6380–6391.

[22] Xianfu Chen, Celimuge Wu, Tao Chen, Honggang Zhang, Zhi Liu, Yan Zhang, Mehdi Bennis. Age of information aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective[J]. IEEE Transactions on Wireless Communications, 2020. 19(4):2268–2281.

[23] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015. 518(7540):529–533.

[24] Shi-Yong Chen, Yang Yu, Qing Da, Jun Tan, Hai-Kuan Huang, Hai-Hong Tang. Stabilizing reinforcement learning in dynamic environment with application to online recommendation[C]//Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. 2018:1187–1196.

[25] Amal Feriani, Ekram Hossain. Single and multi-agent deep reinforcement learning for AI-enabled wireless

networks: A tutorial[J]. IEEE Communications Surveys & Tutorials, 2021. 23(2):1226–1252.

[26] Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P How, John Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability[C]//International Conference on Machine Learning. PMLR, 2017:2681–2690.

[27] Thanh-Dat Le, Georges Kaddoum. A distributed channel access scheme for vehicles in multi-agent V2I systems[J]. IEEE Transactions on Cognitive Communications and Networking, 2020. 6(4):1297–1307.

[28] Le Liang, Hao Ye, Geoffrey Ye Li. Spectrum sharing in vehicular networks based on multi-agent reinforcement learning[J]. IEEE Journal on Selected Areas in Communications, 2019. 37(10):2282–2292.

[29] Hung V Vu, Zheyu Liu, Duy HN Nguyen, Robert Morawski, Tho Le-Ngoc. Multi-agent reinforcement learning for joint channel assignment and power allocation in platoon-based C-V2X systems[J]. arXiv preprint arXiv:2011.04555, 2020.

[30] Alperen Gündoğan, H Murat Gürsu, Volker Pauli, Wolfgang Kellerer. Distributed resource allocation with multi-agent deep reinforcement learning for 5G-V2V communication[C]//Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing. 2020:357–362.

[31] Pablo Hernandez-Leal, Bilal Kartal, Matthew E Taylor. A survey and critique of multi-agent deep reinforcement learning[J]. Autonomous Agents and Multi-Agent Systems, 2019. 33(6):750–797.

[32] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation[J]. Advances in neural information processing systems, 2016. 29:2244–2252.

[33] Jakob N Foerster, Yannis M Assael, Nando de Freitas, Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning[C]//Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016:2145–2153.

[34] Jiechuan Jiang, Zongqing Lu. Learning attentional communication for multi-agent cooperation[C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018:7265–7275.

[35] Daewoo Kim, Sangwoo Moon, David Hostallero, Wan Ju Kang, Taeyoung Lee, Kyunghwan Son, Yung Yi. Learning to schedule communication in multi-agent reinforcement learning[C]//International Conference on Learning Representations. 2018.

[36] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, Joelle Pineau. Tarmac: Targeted multi-agent communication[C]//International Conference on Machine Learning. PMLR, 2019:1538–1546.

[37] Woojun Kim, Jongeui Park, Youngchul Sung. Communication in multi-agent reinforcement learning: Intention sharing[C]//International Conference on Learning Representations. 2020.

[38] Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, Yann Dauphin. On the pitfalls of measuring

emergent communication[C]//Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. 2019:693–701.

[39] Mohamed Salah Zaïem, Etienne Bennequin. Learning to communicate in multi-agent reinforcement learning: A review[J]. arXiv preprint arXiv:1911.05438, 2019.

[40] Tze-Yang Tung, Joan Roig Pujol, Szymon Kobus, Deniz Gunduz. A joint learning and communication framework for multi-agent reinforcement learning over noisy channels[J]. arXiv preprint arXiv:2101.10369, 2021.

[41] Joan S Pujol Roig, Deniz Gündüz. Remote reinforcement learning over a noisy channel[C]//GLOBECOM 2020-2020 IEEE Global Communications Conference. IEEE, 2020:1–6.

[42] Liang Wang, Hao Ye, Le Liang, Geoffrey Ye Li. Learn to compress CSI and allocate resources in vehicular networks[J]. IEEE Transactions on Communications, 2020. 68(6):3640–3653.

[43] Technical Specification Group Radio Access Network. NR; Study on NR Vehicle-to-Everything (V2X)[R]. Valbonne:, Mar. 2019.

[44] Mario H Castaqeda Garcia, Alejandro Molina-Galan, Mate Boban, Javier Gozalvez, Baldomero Coll-Perales, Taylan Şahin, Apostolos Kousaridas. A tutorial on 5G NR V2X communications[J]. IEEE Communications Surveys & Tutorials, 2021. 23(3):1972–2026.

[45] Technical Specification Group Radio Access Network. Overall description of Radio Access Network (RAN) aspects for Vehicle-to-everything (V2X) based on LTE and NR[R]. Valbonne:, Jun. 2020.

[46] Zoraze Ali, Sandra Lagén, Lorenza Giupponi, Richard Rouil. 3GPP NR V2X mode 2: Overview, models and system-level evaluation[J]. IEEE Access, 2021.

[47] 戴子彭 刘驰, 王占健. 深度强化学习 学术前沿与实战应用[M]. 机械工业出版社, 2020.

[48] Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep learning[M]. MIT press, 2016.

[49] Kaiqing Zhang, Zhuoran Yang, Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms[J]. Handbook of Reinforcement Learning and Control, 2021:321–384.

[50] Yaodong Yang, Jun Wang. An overview of multi-agent reinforcement learning from game theoretical perspective[J]. arXiv preprint arXiv:2011.00583, 2020.

[51] Richard S Sutton, Andrew G Barto. Reinforcement Learning, An Introduction[M]. MIT Press, 1998.

[52] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, Daan Wierstra. Continuous control with deep reinforcement learning[C]//ICLR (Poster). 2016.

[53] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

[54] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, Raul Vicente. Multi-agent cooperation and competition with deep reinforcement learning[J]. PloS one,

2017. 12(4):e0172395.

[55] Ryan Lowe, YI WU, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments[J]. Advances in Neural Information Processing Systems, 2017. 30:6379–6390.

[56] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, Yi Wu. The surprising effectiveness of MAPPO in cooperative, multi-agent games[J]. arXiv preprint arXiv:2103.01955, 2021.

[57] Yiping Xing, R. Chandramouli. Stochastic learning solution for distributed discrete power control game in wireless data networks[J]. IEEE/ACM Transactions on Networking, 2008. 16(4):932–944.

[58] Matthew Hausknecht, Peter Stone. Deep recurrent Q-learning for partially observable MDPs[J]. arXiv preprint arXiv:1507.06527, 2015.

[59] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation[C]//Proc. Conf. Empir. Methods Nat. Lang. Process. 2014:1724–1734.

[60] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, Nando Freitas. Dueling network architectures for deep reinforcement learning[C]//International conference on machine learning. PMLR, 2016:1995–2003.

[61] Hado Van Hasselt, Arthur Guez, David Silver. Deep reinforcement learning with double Q-learning[C]//Proceedings of the AAAI conference on artificial intelligence. volume 30. 2016.

[62] Technical Specification Group Radio Access Network. Study LTE-based V2X services; (release 14)[R]. Valbonne:, June 2016.

[63] Juha Meinilä, Pekka Kyösti, Tommi Jämsä, Lassi Hentilä. Winner II channel models[J]. Radio Technologies and Concepts for IMT-Advanced, 2009:39–92.

[64] WhiteGrayxp. MARL-based-Dec-Spectrum-Access. `https://github.com/WhiteGrayxp/MARL-based-Dec-Spectrum-Access`, 2021.

[65] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, David Meger. Deep reinforcement learning that matters[C]//Proceedings of the AAAI conference on artificial intelligence. volume 32. 2018.

[66] Adam Paszke, et al. Pytorch: An imperative style, high-performance deep learning library[J]. arXiv preprint arXiv:1912.01703, 2019.

[67] Maurizio Capra, Beatrice Bussolino, Alberto Marchisio, Guido Masera, Maurizio Martina, Muhammad Shafique. Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead[J]. IEEE Access, 2020. 8:225134–225180.

[68] Peter J Huber. Robust estimation of a location parameter[M]. Springer, 1992.

[69] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, John Schulman. Quantifying generalization in

reinforcement learning[C]//International Conference on Machine Learning. PMLR, 2019:1282–1289.

[70] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, Jun Wang. Mean field multi-agent reinforcement learning[C]//International Conference on Machine Learning. PMLR, 2018:5571–5580.

攻读学位期间的学术论文及研究成果

参与项目：

[1] “城市复杂环境下基于视觉和通信的智能车辆感知与定位，” 国家自然科学基金联合重点项目，2018.1～2021.12。

[2] “2020年工业互联网创新发展工程-‘5G+工业互联网’高质量网络和公共服务平台项目，” 工信部，2020.4～2023.1。

论文：

[1] **P. Xiang**, H. Shan, Z. Zhang, L. Yu and T. Q. S. Quek, “NOMA based VR Video Transmissions Exploiting User Behavioral Coherence,” *Proc. IEEE WCNC*, May 2020, pp. 1-6.

[2] **P. Xiang**, H. Shan, M. Wang, Z. Xiang and Z. Zhu, “Multi-Agent RL Enables Decentralized Spectrum Access in Vehicular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10750-10762, Oct. 2021.

[3] **P. Xiang**, H. Shan, S. Zhou, Z. Zhang and C. Chen, “Multi-agent Reinforcement Learning based Decentralized Spectrum Access in Vehicular Networks with Emergent Communication,” submitted to *IEEE Wireless Communications Letters*.