

基于深度强化学习的车载边缘计算与任务卸载综述^①



雷俊^{1,2}, 刘会衡², 张军², 李慎宏²

¹(湖北文理学院 计算机工程学院, 襄阳 441053)

²(湖北文理学院 物理与电子工程学院, 襄阳 441053)

通信作者: 刘会衡, E-mail: lh@hbuas.edu.cn

摘要: 随着智能交通系统的快速发展, 车联网 (IoV) 对实时计算与低延迟服务的需求激增. 车载边缘计算 (VEC) 通过将任务卸载至边缘节点显著降低了传输延迟. 然而, 传统算法在复杂动态交通环境下对任务卸载适应性不足. 深度强化学习 (DRL) 拥有处理复杂任务的能力, 能够为车辆在复杂动态环境下学习最优的卸载策略. 首先梳理车联网架构、通信技术及 VEC 的核心卸载技术. 其次介绍了 DRL 的基础理论、方法分类以及多智能体协作机制. 然后, 从车-车、车-边缘层和云-边-端资源协同计算卸载这 3 个维度, 全面综述了国内外研究现状. 最后, 展望了基于深度强化学习的车载边缘计算与任务卸载未来可能研究的方向.

关键词: 车联网; 边缘计算; 任务卸载; 深度强化学习

引用格式: 雷俊,刘会衡,张军,李慎宏.基于深度强化学习的车载边缘计算与任务卸载综述.计算机系统应用,2025,34(11):1-19. <http://www.c-s-a.org.cn/1003-3254/10002.html>

Review of Vehicular Edge Computing and Task Offloading Based on Deep Reinforcement Learning

LEI Jun^{1,2}, LIU Hui-Heng², ZHANG Jun², LI Shen-Hong²

¹(School of Computer Engineering, Hubei University of Arts and Science, Xiangyang 441053, China)

²(School of Physics and Electronic Engineering, Hubei University of Arts and Science, Xiangyang 441053, China)

Abstract: With the rapid development of intelligent transportation systems, the demand of the Internet of vehicles (IoV) for real-time computation and low-latency services has surged. Vehicular edge computing (VEC) significantly reduces the transmission delay by offloading tasks to edge nodes. However, traditional algorithms are not sufficiently adaptive to task offloading in complex dynamic traffic environments. Deep reinforcement learning (DRL) is capable of handling complex tasks and learning optimal offloading strategies for vehicles in complex dynamic environments. Firstly, this study sorts out the IoV architecture, communication technology, and core offloading technology of VEC. Secondly, it introduces the basic theory of DRL, classification of methods, and the mechanism of multi-intelligence body collaboration. Then, a comprehensive overview of the current research status at home and abroad is given from the vehicle-vehicle, vehicle-edge layer, and cloud-edge-end resource cooperative computing offloading dimensions. Finally, the possible future research directions of deep reinforcement learning-based VEC and task offloading are pointed out.

Key words: Internet of vehicles (IoV); edge computing; task offloading; deep reinforcement learning (DRL)

① 基金项目: 湖北省自然科学基金 (2025AFD066, 2023AFC002); 湖北省教育厅科学研究计划 (Q20232603)

收稿时间: 2025-04-14; 修改时间: 2025-05-07; 采用时间: 2025-05-26; csa 在线出版时间: 2025-09-30

CNKI 网络首发时间: 2025-10-10

随着智能交通系统和自动驾驶技术的快速发展,车辆也变得智能化,并在摄像头、嵌入式系统、传感器和计算能力方面不断发展,汽车行业越来越多具有强大车载计算能力的车辆被制造出来^[1].这些车辆能支持许多新型车载应用,例如自动驾驶辅助系统^[2]、人工智能、图像辅助导航、增强现实(AR)^[3]、游戏传媒等应用.这些应用需要大量的计算资源,尤其是在高速、复杂的交通环境中,因此资源受限的车辆无法短时间内高效地完成计算密集型和对延迟敏感性的任务^[4].

云计算^[5]具备超大的规模,能整合大量计算和存储资源.将车辆的计算任务卸载至远程云服务器进行处理是一种有效解决方案^[6].虽然传统的云计算模式可以提供强大的计算能力,但是车与云端的距离较远,车辆将任务卸载至云端会导致较长的传输时延和较大的通信开销.

为了弥补这一问题,欧盟标准化协会(ETSI)在2014年首次提出移动边缘计算(mobile edge computing, MEC)的白皮书^[7].MEC是一种将计算、存储和应用功能从传统云端移动到网络边缘的计算架构,它能够有效缓解云计算压力,因此将车辆、任务卸载到MEC可以减轻云计算的压力,减少延迟.

车联网技术的出现最早可追溯到2000年初期,当时主要关注车辆之间的通信(vehicle to vehicle, V2V)、车辆与路边基础设施的通信(vehicle to infrastructure, V2I).随着MEC概念的提出,车载边缘计算(vehicular edge computing, VEC)作为边缘计算的一种应用也被逐步提出,VEC是一个高度分布式和协作的计算架构,由车载设备、边缘计算节点、路边单元(RSU)、云和通信网络等组成^[8].当边缘设备计算资源有限时,需要卸载的车辆可以通过V2V通信,将周围空闲车辆资源利用起来,从而降低边缘设备的压力.

车载边缘计算系统通常涉及大量的车载终端设备和路侧基础设施节点,它们通过无线和有线通信链路进行通信,具有不同处理和延迟要求的应用程序,因此网络所面临的总负载会随时间波动.并且网络还必须高效地应对车辆的移动性,因此,这些场景的复杂性和动态性使得资源管理、决定哪些任务要卸载以及卸载到何处变得更加具有挑战性.尽管在简单、低动态的场景中可以使用传统算法找到解决方案,但它们通常难以在复杂、高动态和多变的车联网环境中实现高效调度.而深度强化学习作为一种基于自适应学习的方

法,适用于更复杂的决策问题,尤其是在车联网这种动态、复杂的环境中,表现出更强的灵活性和智能决策能力^[9,10].

1 车联网

车联网的概念源于物联网,即车辆物联网,是以行驶中的车辆为信息感知对象,借助新一代信息通信技术,实现车与X(车、人、路、服务平台)之间的网络连接,提升车辆整体的智能驾驶水平,为用户提供安全、舒适、智能、高效的驾驶感受与交通服务,同时提高交通运行效率,提升社会交通服务的智能化水平^[11].

1.1 车联网网络架构

车联网技术是在交通基础设施日益完善和车辆管理难度不断加大的背景下被提出的,到目前为止仍处于初步的研究探索阶段,但经过多年的发展,当前已基本形成了一套比较稳定的车联网技术体系结构.在车联网体系结构中,主要由3大层次结构组成,按照其层次由高到低分别是感知层、网络层和应用层^[12].

(1) 感知层

在车联网中,感知层主要用来收集车内外行驶状态信息,道路与交通情况.同时感知层会将采集的信息反馈给网络层和驾驶员,驾驶员或者车辆根据收到的反馈信息进行驾驶决策,可以增加车辆行驶的安全性.

(2) 网络层

车联网网络层是车联网系统中的关键组成部分,负责实现车辆与车辆(V2V)、车辆与基础设施(V2I)、车辆与行人(V2P)以及车辆与网络(V2N)之间的通信^[13].网络层接收感知层传输过来的数据进行分析处理,还能根据网络拓扑和当前的网络状况,选择最佳的传输路径.

(3) 应用层

应用层是车联网架构中的最顶层,直接面向用户和行业需求,负责将底层数据转化为实际服务和应用场景.其核心作用在于整合数据、方法和业务逻辑,为用户、车辆、交通系统等提供智能化和个性化的功能.

1.2 车联网通信技术

随着车联网发展,目前主流的通信技术有2种,分别是C-V2X、DSRC通信技术,其中C-V2X占据着世界车联网发展的主流地位.WAVE架构是一组由IEEE标准化的车载网络通信协议.该架构的目标是为智能

交通系统 (ITS) 提供安全便捷的通信, 并为车辆提供与其他车辆 (V2V) 或基础设施 (V2I) 的直接连接^[14].

(1) C-V2X

2017年, 3GPP第144版LTE标准中首次引用C-V2X. C-V2X是3GPP在LTE-V2X的基础上发展而来的一种车联网通信技术, 该技术使具备C-V2X使用能力的车辆在没有蜂窝基础设施帮助的情况下可以分布式地进行信息交互, 而当车辆在蜂窝网络覆盖内, 可以通过基础设施进行有效的资源分配^[15]. C-V2X有两种主要的通信类型, V2V和V2I. V2V是车辆与车辆之间的通信技术, 在直接传输的情况下, 车辆使用特设通信, 通过WAVE或其他允许D2D通信的技术将任务卸载到其他车辆上. 如果是间接传输, 当路径有几个跳跃点时, 可以使用V2V作为将任务转发到最终目的地的方法. V2I是车辆与基础设施之间的通信, 这种类型的通信允许车辆直接RSU或基站通信, 通过蜂窝、WAVE或其他网络向边缘服务器、微型数据中心、传输云或者其他车辆(通过基础设施)发送加载任务.

(2) DSRC

美国材料实验学会在1992年提出DSRC技术, 后经不断补充完善, 成为IEEE 802.11p的车联网通信标准. DSRC (dedicated short-range communications), 即专用短程通信, 是一种专门为车联网和智能交通系统设计的无线通信技术. DSRC技术对RSU等基础设施存在较强的依赖性, 并且DSRC使用5.850–5.925 GHz的专用频段, 频谱资源相对有限. 这可能导致在高密度交通环境中, 频谱资源不足以支持大量车辆的通信需求, 从而影响通信性能. 与C-V2X相比, DSRC技术在网络容量、高带宽和覆盖范围等方面存在明显劣势.

1.3 车载边缘计算与任务卸载

1.3.1 VEC架构

VEC架构由远程云、边缘云和车辆云这3层结构组成, VEC网络架构如图1所示.

(1) 远程云 (remote cloud, RC): 远程云一般是由云服务器构成的大型服务集群, 具有超强的计算能力和丰富的存储资源, 车辆可以将计算密集型任务卸载到RC. RC凭借自身大规模的计算集群, 能够快速高效地处理任务, 然后将结果返回车辆. 但RC通常部署的位置离车辆较远, 将车辆计算任务传输到RC并得到相应返回结果需要的时间和延迟较高, 对于一些关键任务或具有超低延迟要求的任务可能不适用.

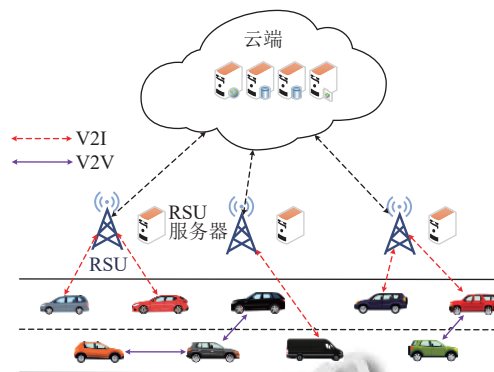


图1 VEC网络架构

(2) 边缘云 (edge cloud, EC): 边缘云主要是由路边单元 (road side unit, RSU)、基站以及移动设备等具备计算、存储和通信能力的边缘设备组成. EC一般部署在车辆和远程云之间, 与车辆相比, EC具有相对丰富的计算资源、存储和通信能力. EC可以直接给车辆提供计算任务卸载服务, 也可以将车辆的计算任务通过回程链路上传到RC. 由于EC距离车辆更近, 车辆到EC的传输时间和通信开销大大降低, 因此在传输时间和通信开销方面EC比RC具有更大的优势.

(3) 车辆云 (vehicular cloud, VC): VC主要由一些未被充分利用或闲置的行驶中车辆和停放车辆组成. 随着汽车工业技术的发展, 现在的汽车已经具有更多的计算和存储能力, 车辆可以实现任务的本地处理. 车辆也可以将计算密集型任务通过V2V卸载到其他有空闲计算资源的车辆上进行处理. 从而减轻边缘云和远程云的计算压力.

1.3.2 计算卸载

计算卸载是一种将计算任务从资源受限的终端设备转移到具有更强计算能力的设备上执行的技术, 该技术可以节省本地计算资源和减少处理大量任务所需的时间. 计算卸载可以分为本地卸载、部分卸载、完全卸载.

(1) 本地卸载: 全部计算任务都由移动终端设备本地处理.

(2) 部分卸载: 部分计算任务由移动终端设备处理, 其余计算任务会卸载到边缘云和远程云, 由边缘服务器和云服务器处理.

(3) 完全卸载: 全部计算任务都卸载到边缘云或远程云处理.

车辆面向边缘服务器计算卸载的流程主要分为6个步骤: 可卸载节点感知、任务类型划分、任务卸载

决策、计算任务传输、计算任务处理、计算结果返回。
计算卸载流程如图 2 所示。

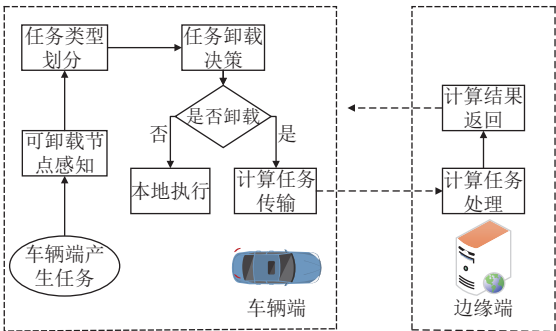


图 2 计算卸载流程

1.3.3 VEC 卸载决策

VEC 卸载决策主要分为集中决策和分布式决策。集中决策是在某个通信区域,由单一管理节点(网络控制器、边缘或者云服务器)对该区域的服务请求车辆做出卸载决策和计算资源分配。集中决策的特点是该管理节点对整个区域的通信条件、服务器和车辆的资源使用情况有全面了解。

分布式决策是车辆自行决定任务是在本地执行还是卸载,无需依赖管理节点进行决策,但分布式决策面临的挑战在于,车辆无法了解其他车辆做出的决策,可能导致边缘节点的通信信道和资源出现拥堵。

1.3.4 VEC 主要卸载技术

VEC 主要卸载技术包括博弈论、启发式算法、契约理论和深度强化学习。

(1) 博弈论: 核心思想是在特定环境下去分析和预测不同参与者选择的策略以及这些策略相互影响的结果^[16-18]。在 VEC 计算卸载中,博弈论主要解决资源分配问题。通过博弈论的理论框架去分析车辆与边缘服

务器之间的相互竞争与合作关系,以提高系统卸载效率。博弈论主要有合作博弈和非合作博弈,在合作博弈中,参与者之间有合作也有竞争,并且在合作中会达到均衡,使得整体利益最大化。而在非合作博弈中,每个参与者独立决策,目标是最大化自身利益。

(2) 启发式算法: 是一类用于解决复杂问题的算法,通过利用问题的某些特征和经验规则,在可接受时间范围内找到较好的近似解。在可接受的时间内,启发式算法虽然不能保证找到最优解,但是可以找到一个较好的解^[19]。常见的启发式算法有模拟退火^[20]、遗传算法^[21]、粒子群算法^[22]、蚁群算法^[23]。

(3) 契约理论: 主要研究如何设计有效的合同,以保障各方的合作,同时维护每个参与者的利益^[24]。在计算卸载领域,该理论可用于设计合适的协议或合同,以促进边缘服务器的运营商与用户之间的合作。例如,在服务设计合同方面,依据用户需求和预期,借助灵活协议来分配计算资源,使用户能挑选适合自身需求的合同,从而保障用户自身利益。当用户和服务器之间有信息不完全或不对称的情况,这时就可以借助契约理论来设计合理的协议,从而帮助双方在真实信息下做出最优决策,以确保合作的有效性和公平性^[25]。

(4) 深度强化学习 (deep reinforcement learning, DRL): 是一种结合了强化学习 (reinforcement learning, RL) 和深度学习 (deep learning, DL) 的机器学习方法,通过深度神经网络来逼近强化学习中的价值函数或策略,从而处理高维复杂环境中的决策问题。深度强化学习方法能够依据车辆端、RSU 和 MEC 服务器的计算能力,然后结合车辆速度、网络带宽、任务负载、能耗状态等信息,动态优化车辆端或服务器的计算卸载决策。VEC 主要卸载技术优缺点对比如表 1 所示。

表 1 VEC 主要卸载技术优缺点对比

技术	优点	缺点
博弈论	能够促使多方参与者之间达到均衡状态,提高系统的稳定性和资源分配效率	对车辆高速移动、边缘服务器资源波动、网络状态变化等动态环境下适应性较弱
启发式算法	能在较短的时间内找到合适的解决方案,适用于需要快速响应的问题。基于经验的规则减少搜索空间,提高了效率	无法提供全局最优保证,对参数选择敏感,且难以应对复杂的动态环境
契约理论	适用于信息不对称环境,通过设计合适的契约保障各方利益	缺乏灵活性,在动态决策时较为困难,通常依赖于固定协议
深度强化学习	具备自适应学习能力和动态优化能力,能够有效应对复杂且不断变化的环境	训练复杂度高,需要大量数据和计算资源。奖励函数的设计直接影响智能体行为,若设计不当会导致意外结果

2 深度强化学习

2.1 基本原理

在强化学习中,智能体通过在离散时间步长内依

次选择动作,并与一个随机且完全可观测的环境进行交互,以最大化累计奖励,这一系列过程被称为马尔可夫决策过程 (Markov decision process, MDP)^[26]。

MDP 可以表示为一个五元组 (S, A, P, R, γ) . 其中, S 表示所有环境状态的集合; A 是智能体可执行动作的集合, 可以是离散的或连续的; P 是满足马尔可夫决策性质的转移概率函数, 表示为 $S \times A \rightarrow \pi(s)$; $\gamma \in [0, 1]$ 是折扣因子, 主要是用来平衡未来奖励对当前奖励的影响, 防止累计奖励发散. 在 MDP 中, RL 的任务目标是找到最大化累计奖励. 累计奖励是从时间 t 开始的所有折扣奖励之和.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (1)$$

其中, G_t 表示长期回报, R_t 表示第 i 个时间步的即时奖励.

1950 年, Bellman^[27] 提出贝尔曼方程, 它通过递归的方式, 将一个问题的最优解转化为子问题的最优解, 从而通过分治法来求解复杂问题. 贝尔曼方程在 MDP 中有广泛的应用, 主要用来计算最优策略或者最优值函数.

(1) 状态值函数 $v_{\pi}(s)$: 在给定策略 π 下, 从状态 s 开始所能获得的期望回报.

$$v_{\pi}(s) = \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) [r + \gamma v_{\pi}(s')] \quad (2)$$

其中, $\pi(a|s)$ 即为策略 $\pi(a_t|s_t)$, 表示智能体在状态 s_t 下选择 a_t 的概率, 在执行完该动作后, 根据状态转移函数 $p(s', r|s, a)$ 转移到下一个状态 s_{t+1} 并获得即时奖励 r , 这个过程会递归进行到结束, 最终得到期望回报. 为最大化期望回报 $v^*(s)$, 需要找到在状态 s_t 的期望回报最大的最优策略 $\pi^*(a_t|s_t)$.

$$v^*(s) = \max_a \sum_{s'} p(s', r|s, a) [r + \gamma v^*(s')] \quad (3)$$

(2) 动作值函数 $q^{\pi}(s, a)$: 在给定状态 s 下采取动作 a 后, 遵循策略 π 所能获得的预期回报.

$$q^{\pi}(s, a) = \sum_{s', r} p(s', r|s, a) [r + \gamma v^{\pi}(s')] \quad (4)$$

最优动作值函数 $q^*(s, a)$ 是每个状态 s 和动作 a 的最大值, 最优策略 $\pi^*(s)$ 选择的动作 a 是最优动作 $a^*(s)$.

$$q^*(s, a) = \sum_{s', r} p(s', r|s, a) [r + \gamma \max_{a'} q^*(s', a')] \quad (5)$$

$$\pi^*(s) = \arg \max_a q^*(s, a) \quad (6)$$

2.2 深度强化学习主要方法

本节对深度强化学习主要方法进行分类. MDP 是

深度强化学习的核心结构, 大多数的深度强化学习方法都是通过 MDP 实现, 这类的方法主要分为基于模型和无模型两类. 基于模型的深度强化学习的核心思想是学习一个可以进行决策的模型, 从而能够在不直接与环境交互的情况下模拟环境. 基于无模型的深度强化学习则通过与环境的直接交互来学习最优策略或值函数.

2.2.1 基于模型的深度强化学习方法

基于模型的深度强化学习方法可以分为学习模型和给定模型.

(1) 学习模型

学习模型核心思想是智能体能够在已经学习好的环境模型中进行决策, 这样大大减少与真实环境的交互. I2A 是一种反向推理的模型学习方法, World Model 是通过神经网络学习的世界模型, 用于环境模拟和决策规划. 这两种方法归于学习模型.

(2) 给定模型

给定模型核心思想是智能体直接使用已知的环境模型进行决策, 无需与环境进行交互学习. 蒙特卡洛树搜索 (MCTS) 结合了蒙特卡洛随机模拟和树搜索, 用于决策过程中的搜索和优化, 主要应用在 AlphaGo 和 AlphaZero. MCTS 的模型是在特定场景训练过的给定模型.

2.2.2 基于无模型的深度强化学习方法

无模型的深度强化学习方法主要分为基于值函数的深度强化学习和基于策略梯度的深度强化学习, 以及结合两者优势的 Actor-Critic^[28] 框架.

蒙特卡洛 (MC)^[29] 和时间差分 (TD)^[30] 是强化学习的主要方法, 也是无模型方法的基石. MC 方法通过与环境交互获得完整经验轨迹来计算回报的均值, 进而估计状态或动作的状态值函数或动作值函数. 其优势是直观且无偏的, 但高方差和依赖完整序列的特性, 使其在部分场景下效率较低. TD 结合 MC 的经验平均方法和动态规划的自举法, 直接利用环境交互经验在线更新价值函数.

(1) 基于值函数的深度强化学习

基于价值的深度强化学习通过直接学习状态或状态-动作对的价值函数来间接优化策略. 它需要为每个状态-动作对估计 Q 值, 最优策略则是为每个状态选择具有最大 Q 值的动作. 通过最小化时间差分 (TD), 可以迭代地估计 Q 值^[31]. 基于值函数的深度强化学习方

法分为在线策略 (on-policy) 和离线策略 (off-policy), 在线策略是智能体与环境实际交互, 并根据获得的数据进行学习 and 优化策略; 离线策略则是智能体利用已经存储的数据 (例如经验回放) 来更新策略, 从而减少对环境的交互需求. Q-learning^[32]方法和 SARSA^[33]方法把每个状态或者状态-价值函数存储在表格中. 当状态空间或者动作空间较大时, 用表格来存储和计算

Q 值会变得困难. 为解决该困难, 使用深度神经网络来近似 Q 值函数, 从而代替传统的 Q 表格存储. 典型方法包括深度 Q 网络 (DQN)^[34]及其变种双 DQN (Double DQN)^[35]、对战 DQN (Dueling DQN)^[36]、彩虹 DQN (Rainbow DQN)^[37]. 除了 Q-learning 方法是在线策略, 其他方法都是离线策略. 基于值函数的主要方法对比如表 2 所示.

表 2 基于值函数主要方法对比

方法	优点	缺点	适用场景
Q-learning	不需要模型, 可以直接和环境交互学习. 还可以采用离线学习的方式, 更新Q值不依赖实时交互	移动车辆在进行任务卸载时状态空间和动作空间较大, 然而Q表存储空间有限, 容易出现维度灾难	适用于简单的任务卸载决策场景, 如静态环境下的单车任务分配. 然而, Q-learning不适合处理大规模状态空间、高动态环境或奖励信号稀疏的任务卸载场景
SARSA	是在线学习方法, 无需事先了解环境模型. 通过与环境的实时交互学习最优策略. 相比Q-learning更稳定	收敛速度较慢, 探索能力较弱. 如果环境有多个相似状态, 该方法可能会陷入局部最优解	适用于动态变化较小、任务卸载决策简单的场景. 然而, 在面对复杂、高动态的环境、奖励信号稀疏的场景时, 可能会遇到学习效率低、策略更新不稳定的问题
DQN	能够处理高维连续状态和动作空间, 引入经验回放机制, 能很好地处理延迟和奖励问题	易出现过估计问题, 需要较长时间的训练才能达到较好效果	适用于高维观测空间和复杂环境下的任务卸载场景. 不适合用于高实时性要求、奖励信号稀疏或过度依赖探索的场景
Double DQN	有两个Q网络, 减少过估计问题, 在车联网环境中, 相比DQN有更快的收敛速度和学习效率	依赖两个Q网络的同步更新, 在训练时候, 如果更新不同步易出现较大方差	适用于中等动态、中等稀疏奖励的车载环境, 能通过减少Q值过估计提升策略稳定性. 但不适用高动态或奖励信号稀疏场景中
Dueling DQN	将状态值和优势函数分开建模, 相比DQN减少计算复杂度. 能从有限的中提取更泛化性的决策模式, 使任务卸载更稳定	相比DQN结构更复杂, 增加了网络的复杂度和训练时间, 并且超参数设置更加敏感	适用于需要稳定策略的任务卸载, 松散耦合的多任务卸载, 资源有限的车载边缘计算任务. 在高动态变化的环境、奖励稀疏的任务卸载场景, 该方法性能可能会受到限制
Rainbow DQN	主要集成了Double DQN和Dueling DQN的优点, 能够在较少训练步数下达到较好的性能, 能适应复杂多变的车辆环境	实现比较复杂, 计算资源需求大, 训练时间较长和超参数众多调整复杂	适用于高度动态的车联网场景、计算资源充足的边缘服务器长期规划任务和离散化任务决策. 不适用连续动作空间决策、奖励信号稀疏场景和高实时决策
CS1	通过学习回报而非单一Q值, 能够捕捉回报的波动性, 从而提高价值估计的精准度	相比DQN的不同, CS1的网络输出不是行为值函数而是支点处概率	适用于回报不确定性和奖励波动较大的任务卸载决策场景. 不适用于奖励信号稳定、低资源设备和高实时性要求的场景

(2) 基于策略梯度的深度强化学习

核心思想是直接优化策略网络参数, 通过最大化期望累计奖励的梯度来更新策略, 而非依赖值函数的间接估计. 该方法利用深度神经网络参数化策略, 生成动作概率分布, 并通过梯度上升法调整参数, 使策略在环境中选择的动作能获得更高长期回报. 基于策略的强化学习方法分为基于梯度和无梯度两种. 基于梯度的代表性方法, 有蒙特卡洛策略梯度 (REINFORCE)^[38]、近端策略优化 (PPO)^[39]、信任域策略优化 (TRPO)^[40]、自适应策略梯度算法 (ACKTP)^[41]. 基于无梯度的代表性方法, 包括基于在线优化的 Q 迁移方法 (QT-Opt) 和 SAMUEL. 基于策略函数主要方法对比如表 3

所示.

(3) 基于 Actor-Critic 框架的深度强化学习方法

Actor-Critic (AC) 框架结合了值函数 (Critic) 和策略网络 (Actor), 通过 Critic 评估动作价值并指导 Actor 更新, 典型包括深度确定性策略梯度 (DDPG)^[42]、双延迟深度确定性策略梯度 (TD3)^[43]和软 Actor-Critic (SAC)^[44]、异步优势执行者-评估者 (A3C)^[45]、优势执行者-评估者 (A2C)^[46]. 基于 Actor-Critic 框架主要方法对比如表 4 所示.

本节主要参考了深度强化学习相关综述^[47-53], 文献^[54,55]主要讲解了深度强化学习相关原理. 深度强化学习主要方法分类如图 3 所示.

表3 基于策略梯度主要方法对比

方法	优点	缺点	适用场景
PPO	数据利用高效, 训练稳定性好. 在车载动态环境下, 能保障任务卸载和资源分配的有效更新	在车辆任务卸载高维状态动作空间需要大量采样, 导致训练时间较长, 采样不充分会陷入局部最优	适用于连续动作空间的任务卸载. 不适用于离散动作空间、奖励信号稀疏和高实时性要求的任务卸载场景
TRPO	相比PPO, 该方法在奖励稀疏环境下训练稳定性更好	相比PPO, 计算复杂度高, 需要采样更多的样本, 训练时将更长	适用于策略更新频率低和奖励信号稀疏场景. 不适用于高实时性和简单任务卸载场景
REINFORCE	原理简单易实现, 适用于简单离散的任务卸载决策场景	在高维的状态动作空间训练不稳定. 每次更新策略时需要等待整个轨迹的回报收敛速度较慢	适用于离散动作小、状态空间简单的任务卸载场景, 不适用高维状态空间、奖励信号稀疏或高实时性要求的任务卸载决策场景
ACKTP	相比PPO/TRPO, 该方法采用效率和训练稳定性更高	内存占用较大, 不适用于内存受限的设备上, 训练时间较长	适用于连续和离散动作空间、稳定性要求高的场景. 不适用高实时性和资源受限的场景

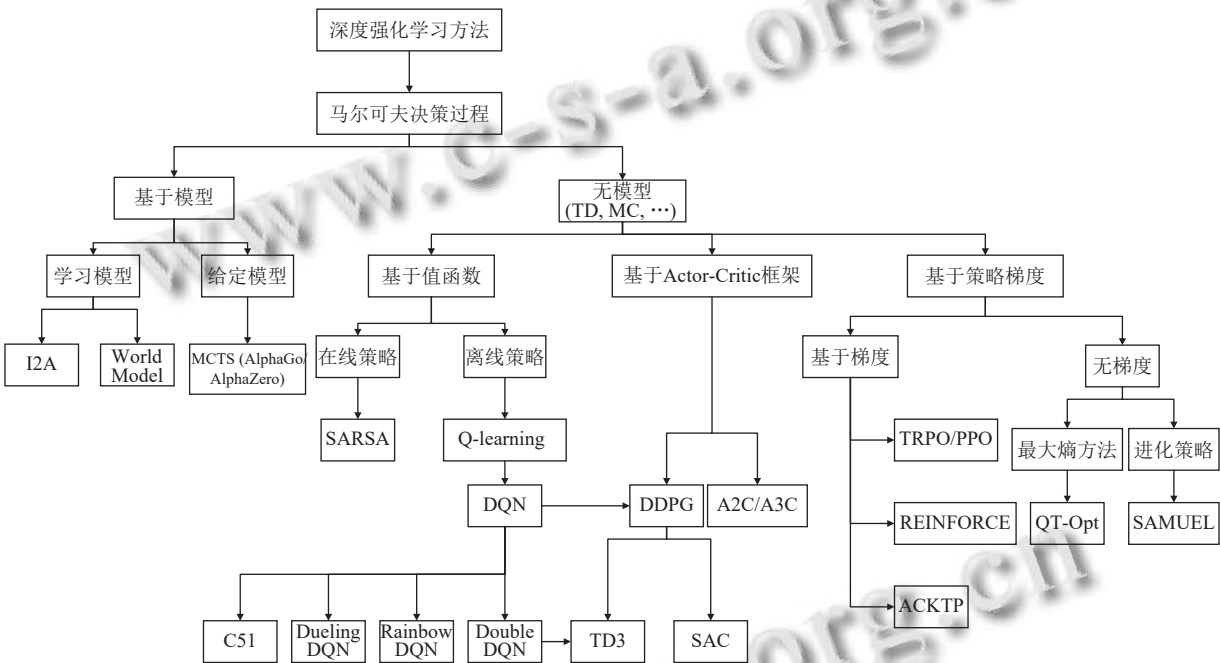


表4 基于 Actor-Critic 框架主要方法对比

方法	优点	缺点	适用场景
DDPG	连续动作空间处理能力强, 稳定性较好, 收敛速度快. 适用于资源分配等	对超参数敏感, 在车载环境中的网络波动等因素下, 策略震荡难以收敛	适用于需要连续控制场景. 不适合离散动作空间、奖励信号稀疏或计算资源受限的场景
TD3	对目标网络采用延迟更新, 提升了训练稳定性. 通过最小化两个Q值来估计动作价值, 减少过估计偏差	超参数数量增多且互相影响, 调优难度大. 相比于DDPG, 增加了网络结构和计算量. 对资源受限的车辆端不适合	适用于连续动作空间和稳定性要求高的任务卸载场景. 不适用离散动作空间、计算资源受限和奖励信号稀疏的场景
SAC	在优化策略时考虑熵项, 智能体能探索更多动作, 避免陷入局部最优. 能够批量采样, 提高训练效率	有两个Q网络较为复杂, 对车载设备的计算资源要求高. 超参数敏感, 增加了调整参数的难度	适用于复杂多变的动态环境和任务卸载决策需要充分探索策略的场景. 不适用离散动作空间、计算资源有限的设备和奖励信号稀疏的场景
A2C	结构比A3C简洁, 计算量较小, 更新稳定性更强	策略梯度估计方差较大, 训练不稳定	适用于状态空间简单的任务卸载场景. 不适用奖励信号稀疏、高维状态空间和高实时性任务卸载的场景
A3C	采用异步训练的方法, 能从多个智能体并行收集样本, 提高样本利用率	异步更新机制导致稳定性较差, 并行训练设计多个参数设置, 超参数调整难	适用于连续动作空间和实时任务卸载决策场景. 适用于离散动作空间和计算资源受限的场景

2.3 多智能体深度强化学习

多智能体深度强化学习 (multi-agent deep reinforcement learning, MADRL) 分支. MADRL 主要研究多个智能体在信息共享环境中进行学习和决策, 每个智能体的动作都会影响其他智能体的奖励, 通常采用纳什均衡来判断所有智能体是否达到均衡状态. 在单智能体强化学习中, 智能体与环境的交互遵循马尔可夫决策过程, 目标是找到一个最优的策略, 让智能体在与环境的交互中获得最大的长期回报. 本节将简要介绍 MADRL 的 3 种学习范式和多智能体系统中智能体的 3 种竞争关系^[56-61].

2.3.1 多智能体深度强化学习的范式

MADRL 的 3 种学习范式主要分为独立学习、集中学习、集中式训练分布式执行.

(1) 在独立学习范式中, 每个智能体都独立地优化各自的行为策略. 每个智能体的行为都会当作环境的一部分, 从而导致在多智能体环境中无法有效处理智能体之间的协作.

(2) 在集中学习范式中, 所有智能体的学习过程集中到一个中心系统或一个控制器上, 中央系统通过获得所有智能体的状态信息、动作选择和奖励反馈来进行决策优化. 但是智能体的状态空间和动作空间会随着智能体数量的增加而呈指数级增长, 从而产生维数灾难.

(3) 在集中式训练分布式执行范式中, 智能体系统采用中心 DRL 模型协同训练机制. 在训练阶段, 各智能体可以通过交互获取彼此的信息, 将其交互过程中获取的信息输入到中央训练模块进行训练. 当完成模型收敛后, 系统转入分布式运行模式, 此时每个智能体基于去中心化决策机制, 仅需要根据本地环境观测数据与相邻节点的实时状态反馈, 即可通过训练策略网络独立生成最优动作指令.

基于值函数分解方法和基于中心值函数方法都是采用集中训练分布式执行范式. 基于值函数分解方法的核心思想是将全局联合值函数拆解为各个智能体的个体值函数, 从而让每个智能体能够独立决策, 同时保证整体策略. 文献^[62]根据是否引入其他机制以及引入机制的不同将基于值分解的多智能体深度强化学习方法分为 3 类: 简单因子分解型、基于 IGM (个体-全局-最大) 原则型以及基于注意力机制型; 然后按分类重点介绍了几种典型方法并对方法的优缺点

点进行对比分析; 最后简要阐述了所提方法的应用和发展前景. 基于中心值函数方法的核心思想是使用全局信息训练价值网络, 通过价值网络指导每个策略网络独立学习.

2.3.2 智能体之间的竞争关系

在多智能体环境中, 智能体之间存在完全竞争关系、完全合作关系、部分合作部分竞争的混合关系. 在完全竞争关系的多智能体环境下, 智能体的目标是彼此对立的, 每个智能体的目标是最大化自己的收益. 此类环境中的策略设计通常依赖于博弈论, 且智能体行为高度相互依赖, 导致学习过程充满竞争和不确定性. 在完全竞争的关系下, 一般使用 Minimax-Q 方法^[63], 该方法可以找到多智能体强化学习的纳什均衡, 但它属于对手独立方法, 当对手未采用纳什均衡策略, 而是使用较差策略时, 该方法无法让智能体依据对手策略调整优化自身策略, 只能得到随机博弈的纳什均衡策略, 即便对手策略很弱, 当前智能体也无法学习到比纳什均衡策略更好的策略. 在完全合作关系中, 所有智能体具有相同的奖励函数, 目标是最大化整体团队收益, 而不是各自独立优化. 在部分竞争部分合作的混合关系中, 主要分为智能体独立方法和智能体感知方法, 智能体独立方法一般采用基于 Q 学习的通用结构, 其中策略值和状态值使用 SG 中的博弈论求解器^[64]计算. 而感知方法使得智能体在决策时考虑其他智能体的行为或状态, 从而使自身的策略达到最优.

3 车载边缘计算与任务卸载

本节从基于车-车的计算卸载、基于车-边缘层的计算卸载、基于资源协同的计算卸载这 3 个维度对国内外研究现状进行总结. 车载计算卸载的主要优化目标有延迟优化、能耗优化、服务质量 (QoS) 优化、系统负载均衡等.

3.1 基于车-车计算卸载

随着车联网的发展, 现在的车联网是由安装传感器、计算单元和软件的移动车辆组成^[65]. 车辆的车载计算单元能够支持延迟敏感和计算密集型的应用, 使得车辆可以通过边缘服务器卸载任务, 从而减轻基础设施节点的负载^[66].

文献^[67]考虑了一个由一辆装有多多个应用程序的车辆和多辆可用于卸载任务的车辆组成的车载网络. 作者考虑了移动车辆将任务卸载到其他车辆上时的 3

种情况, 分别是任务传输成功、部分任务传输成功和任务传输失败. 为了最小化车辆卸载延迟和能耗的系统成本, 作者提出一种基于贝叶斯推理集成的 DQN 解决方法, 通过运用该方法和全部本地计算、全部边缘服务器计算和随机计算进行对比, 在任务数量较少时 (<10), 除随机计算方案外, 其他方案的总消耗成本较为接近; 当任务数量在 10–20 之间, 该方法明显优于其他方法. 文献[68]提出 CORADQN, 通过将计算密集型任务拆分为多个子任务, 充分利用周围空闲服务车辆的计算资源, 将计算任务并发卸载至 MEC 服务器、空闲服务车辆和本地车辆进行处理. 作者用该方法和全部本地计算、全部边缘服务器计算、Fos 方法和 DQN 方法进行对比, 随着数据块大小 (200–1 200 KB) 的增加系统总延时小于其他方法. 当数据块在 1 200 KB 时可以分别减少 78%、64.1%、53.2% 以及 26.7% 的系统总延迟. 文献[69]提出基于 DQN 改进的 DODQN 方法. 在高速公路的场景下, 作者将车辆的计算密集型任

务拆分为多个子任务. 并将子任务分配到多个服务器和车辆上, 经过与部分淹没算法 (PFA) 的对比实验, 该方法在减少卸载延迟方面具有性能提升. 文献[70]考虑了计算资源的时空异质特性并利用周围存在的闲置计算资源. 为了最小化任务时间, 作者提出一种基于分层架构和反事实多智能体强化学习 (COMA) 的分布式任务卸载方法, 传统方法多局限于单一区域, 导致资源利用效率低下, 作者通过两层架构实现跨区域协同: 底层由路边单元 (RSU) 管理本地服务区域, 上层将 RSU 分组形成协作组以平衡全局资源. 为解决多智能体协作中的信用分配问题, 采用 COMA 方法, 结合集中式训练与分布式执行, 通过反事实基线评估个体动作对全局奖励的贡献, 优化任务卸载决策. 作者将该方法与 NF-max、COMA-S 和 RBA 方法进行了对比, 全局服务延迟随着服务车辆数量增加 (5–45), 全局服务延迟明显下降且优于其他方法, 初期下降趋势显著, 后期趋于平缓. 车-车计算卸载主要算法比较如表 5 所示.

表 5 基于车-车计算卸载算法比较

探索方法	基础算法	优化目标	卸载模型	决策方式	车辆移动性	任务依赖性
文献[67]	DQN	最小化卸载延迟和能耗	部分卸载	集中决策	移动	依赖
文献[68]	CORADQN	最小化总任务执行时间	部分卸载	分布式决策	移动	独立
文献[69]	DODQN	最小化系统总延迟	部分卸载	集中决策	移动	独立
文献[70]	COMA	最小化任务执行时间	部分卸载	分布式卸载	移动	独立

3.2 基于车-边缘层计算卸载

当车辆无法单独完成计算密集型和复杂任务时, 车辆可以通过 V2I 将任务卸载到具有强化计算能力的路边单元或 MEC 服务器上进行处理, 从而有效缓解本地计算资源的压力、减少延迟^[71].

3.2.1 降低延迟和能耗

文献[72]考虑车辆行驶在有多个边缘服务器的路上, 为了选择最佳服务器来卸载任务, 作者提出两种方法: 一种是离线方法 (ORLO), 在开始行驶前选择最佳候选服务器; 另一种是基于 DQN 的在线方法, 根据网络条件从候选服务器集中选择最优服务器. 作者对这两种方法进行对比, 低容忍时延 (500–800 ms) 下, ORLO 任务卸载成功率高于 DQN 达 10%–15%, 中高容忍时延 (800–1 500 ms) 下, ORLO 仍领先 5%–10%. 文献[73]考虑有多个基站的超密集网络中的边缘系统, 还考虑了车辆的移速. 作者设计了一种跟随计算卸载范式, 其中移动服务器可以提供额外的计算资源, 使用深度 Q 网络作为计算卸载策略以降低延迟和能耗. 作者将

全部固定 MEC 服务器卸载、全部移动服务器卸载和 FCO-DQN 方案动态选择固定或移动服务器卸载, 实验表明在速度差较小或计算任务量大而 CPU 频率较小的情况选择移动服务器优于固定服务器, 并且 FCO-DQN 方案有效降低了延迟和能耗. 文献[74]考虑了一个具有随机任务生成的动态无线环境. 在这一移动车辆的场景中, 作者提出了一种基于 PPO 的方法, 用于决定何时何地卸载, 从而最大降低任务延迟和能耗之间权衡的长期成本. 尽管作者描述了一个具有多层多设备网络架构的场景, 但强化学习方法的动作空间较为有限, 仅考虑了生成任务的设备、一个基站和一个移动边缘计算服务器.

文献[75]提出了一种基于策略梯度 (PG) 的卸载方案, 以最小化在车联网领域中应用程序的延迟. 特别是, 作者考虑了一个具有多个边缘服务器的多个车辆环境. 该方法的优点在于将应用程序划分为具有依赖关系的任务, 这些依赖关系以有向无环图 (DAG) 的形式表示. 作者用该方法和 FCFS、Random 进行对比, 当子任务

数量(5–30)和车辆数量(10–35)增加时该方法明显优于其他两种方法. 文献[76]继续开展相关工作, 考虑了MEC环境下的车载网络, 其中多个车辆和多个MEC服务器连接到不同的路侧单元, 还有一个中央MEC服务器. 此外, 作者还考虑了将应用划分为具有依赖关系的任务, 用有向无环图表示, 并采用基于PG的方法进行计算卸载. 不过, 该方法采用了分布式方法, 每辆车都有一个代理, 其决策基于车辆的任务、已知的网络状态以及其他车辆的过往决策. 而且, 这一次作者将其方案与两种强化学习方法AC和DQN以及其他卸载方案进行了比较, 结果表明其方案具有更高的收敛性和更好的性能. 文献[77]考虑了一个包含数据中心、MEC服务器和车辆的车载边缘网络. 数据中心负责训练模型, 而每辆车都包含一个智能体, 用于选择一个MEC服务器来卸载任务. 作者提出的多智能体深度强化学习(MDRCO)的计算卸载方案, 该方案考虑了多车辆环境的不确定性, 使车辆能够做出卸载决策以获得最优的长期奖励, 也能观察状态下对MEC服务器的最优卸载决策, 从而最小化长期内的总任务处理延迟.

文献[78]考虑多个车和多个边缘服务器的边缘环境下车联网计算任务卸载场景, 作者提出深度强化学习的Actor-Critic方法训练指针网络, 运用训练好的指针网络对车辆卸载任务进行调度, 使任务超时率最小. 文献[79]考虑车辆在连续时间内进行任务卸载, 作者提出一种基于DQN和DDPG结合的分阶决策深度强化学习方法去解决该问题, 随着时间变化(0–300 ms), 该方法总延迟明显优于DQN、DDPG、本地计算和贪婪卸载. 文献[80]考虑高速公路场景, 构建含RSU和车辆的网络架构. 作者考虑了移动车辆网络的动态特性, 提出了DQN方法, 该方法利用经验回放和参数优化, 有效地降低了能耗和延迟.

文献[81]考虑车联网场景中传统多接入边缘计算(MEC)服务器存在的延迟高、能耗大、覆盖不足等问题, 以及动态环境下车辆移动性、信道状态和资源分配的多变特性, 提出一种基于Actor-Critic深度强化学习的方法来解决该问题, 该方法用来确定任务是在本地车辆执行还是卸载到RSU单元执行. 文献[82]提出了一种基于深度强化学习的优先级深度Q网络(DQNP)方法, 该方法考虑任务优先级, 以在所有优先级级别内最大限度地提高任务完成率, 同时最小化延迟和能耗. 实验表明DQNP的平均延迟最低, 约为50.56 ms. 并且

实现了整体任务完成率86.7%, 与DQNB(总体72.7%)和延迟贪心算法(总体74%)相比, 分别有14%和12.7%的显著提升. 文献[83]考虑多服务多移动用户的场景, 将多用户卸载问题分为卸载决策阶段和请求调度阶段, 并设计了一种基于双深度Q网络(DDQN)的奖励评估方法, 该方法考虑了任务优先级和终端的移动特性.

文献[84]考虑车辆速度, 为了最小化能耗, 作者提出基于多智能体深度确定性策略梯度(MADDPG)方法来获取任务卸载和资源分配策略, 该方法根据任务优先级、任务大小、车辆速度以及车辆信道状态制定出最优的卸载和资源分配策略, 并尽快降低所有车辆的能耗. 文献[85]研究了在车载网络中使用部分计算卸载的情况. 作者提出了两种场景, 一种是单车场景, 另一种是多车场景, 在这些场景中, 车辆可以使用A3C方法将其任务部分卸载到MEC服务器上执行. 车-边缘层计算卸载方案中, 基于降低延迟和能耗的主要算法比较如表6所示.

3.2.2 负载均衡

从网络的宏观角度来看, 确保网络上所有设备之间的负载平衡仍然是一个挑战. 因此, 设计方法能充分利用计算资源和网络资源非常重要. 设计的方法对资源管理不当, 会导致部分设备过载, 且许多设备可能闲置. 这将影响网络 and 应用程序的性能^[86].

文献[87]考虑多车多路边单元的车辆边缘计算场景, 为了最小化所有车辆的总任务处理延迟和能耗, 作者提出一种基于确定性策略梯度(TD3)的深度强化学习方法来高效解决该问题, 该方法中嵌入了一个优化子程序, 通过数值方法解决两个子问题, 从而降低了方法的训练复杂度. 文献[88]考虑了具有多个服务器的车载MEC网络. 该MEC环境包含多个连接到基站的车辆. 作者提出使用基于DDPG的方法进行计算卸载, 以最小化应用程序执行过程中的延迟, 并实现本地执行与服务器执行之间的负载均衡. 随着车辆数量(10–80)的增加, 该方法相比全部卸载到边缘服务器和带负载均衡系数的Actor-Critic方法, 该方法的平均卸载延迟增长率最低. 文献[89]考虑了车辆移动过程中, 无线信道和带宽随时间的变化. 为了在能量消耗成本和数据传输延迟成本之间取得平衡, 作者提出一种基于深度强化学习(ACORL)的可处理连续动作空间的自适应计算卸载方法. 文献[90]为了解决因车辆卸载大量任务导致的边缘节点之间的负载不均衡问题, 提出双延迟深度确定

性策略梯度算法 (TD3) 和基于与理想解相似度优先排序技术的服务器选择算法 (TOPSIS), 从而降低系统的成本, 并使各服务器之间负载均衡. 基于车-边缘层计算卸载方案中, 基于负载均衡的主要算法比较如表 7 所示.

表 6 基于降低延迟和能耗方面的工作总结

探索方法	基础算法	优化目标	卸载模型	决策方式	车辆移动性	任务依赖性
文献[72]	DQN	最小化卸载延迟	部分卸载	分布式决策	移动	独立
文献[73]	DQN	降低卸载延迟和能耗	部分卸载	分布式决策	移动	独立
文献[74]	PPO	最大降低任务延迟和能耗的长期成本	完全卸载	分布式决策	移动	独立
文献[75]	PG	最小化所有车辆应用的总任务执行延迟	部分卸载	分布式决策	移动	依赖
文献[76]	PG	最小化多车辆场景下的总任务延迟和能耗	部分卸载	分布式决策	移动	依赖
文献[77]	MDRCO	最小化总任务延迟	完全卸载	分布式决策	移动	独立
文献[78]	Actor-Critic	最小化任务超时率	完全卸载	集中决策	移动	独立
文献[79]	DQN、DDPG	最小化任务执行时延和能量消耗	完全卸载	分布式决策	移动	独立
文献[80]	DQN	最小化受网络资源影响的能源消耗	完全卸载	集中决策	移动	独立
文献[81]	Actor-Critic	最小化系统能量消耗和卸载延迟	完全卸载	集中决策	移动	独立
文献[82]	DQNP	最小化卸载延迟和能耗	完全卸载	集中决策	移动	独立
文献[83]	DDQN	最大化单位时间内完成的重要任务数量	完全卸载	分布式决策	移动	独立
文献[84]	MADDPG	最小化卸载能耗	完全卸载	集中决策	移动	独立
文献[85]	A3C	最小化任务执行延迟	部分卸载	集中决策	移动	独立

表 7 基于负载均衡算法比较

探索方法	基础算法	优化目标	卸载模型	决策方式	车辆移动性	任务依赖性
文献[87]	TD3	RSU负载均衡, 最小化卸载延迟和能耗	完全卸载	分布式决策	移动	独立
文献[88]	DDPG	MEC服务器负载均衡, 最小化卸载的延迟和能耗	完全卸载	集中决策	移动	独立
文献[89]	ACORL	MEC服务器负载均衡, 最小化MEC服务器成本	完全卸载	集中决策	移动	独立
文献[90]	TD3、TOPSIS	MEC服务器负载均衡, 最小化系统成本	部分卸载	集中决策	移动	独立

3.2.3 降低成本

对于用户而言, 最重要的是用户体验, 因而保障优质的服务质量是考察计算卸载技术方案优越性的标准之一. 由于部分终端设备成本较高, 在此过程中, 如何降低服务提供商 (如云服务提供商和边缘服务提供商) 的总成本, 同时最大化利润, 是需要关注的关键问题^[91].

文献[92]考虑车辆的移动性、期限约束和 MEC 服务器资源的有限性, 提出了一种基于深度强化学习方法的深度确定性策略梯度 (DDPG) 来学习 MEC 服务器用户的功率分配策略. 文献[93]考虑了两种典型的 MEC 架构, 即 MeNB 和 EN 挂载的 MEC 服务器,

为减少时延和优化服务质量 (QoS), 作者提出深度确定性策略梯度和分层学习架构. 文献[94]考虑复杂的城市交通网络场景, 提出基于 DDPG 的方法, 以确定车辆的任务卸载、计算和结果交付策略. 通过优化负载分配和服务器选择, 使协同计算中的服务成本最小化. 文献[95]设计了车联网的系统架构. 路边单元接收社区内的车辆数据, 并将其传输至移动边缘计算服务器进行数据分析, 同时控制中心收集所有车辆信息. 该方法以最小化用户成本函数为目标函数, 采用深度强化学习方法中的 DDQN 来解决由用户移动导致的网络状态实时变化问题. 车-边缘层计算卸载方案中, 基于降低成本的主要算法比较如表 8 所示.

表 8 基于降低成本算法比较

探索方法	基础算法	优化目标	卸载模型	决策方式	车辆移动性	任务依赖性
文献[92]	DDPG	最小化系统成本	完全卸载	分布式决策	移动	独立
文献[93]	DDPG	最大化满足车载应用的服务质量	完全卸载	集中决策	移动	独立
文献[94]	DDPG	最小化服务成本	部分卸载	集中决策	移动	独立
文献[95]	DDQN	最小化用户成本	部分卸载	集中决策	移动	独立

3.3 基于云-边-端资源协同计算卸载

云-边-端 3 层架构是通过将云计算、边缘计算和

终端设备相结合的架构. 云-边-端资源协同卸载是车载设备、路边单元、边缘计算服务器和云服务器等多个

计算节点之间的协作,通过利用各自的计算资源和网络资源一起来处理任务卸载和资源分配,提高计算效率、减少延迟、节省能耗,并优化整体系统性能。

文献[96]提出一种基于 Q-learning 的方法,该方法解决了车辆、MEC 服务器和云服务器网络中的计算卸载和资源分配问题。该方法可以将车辆的计算任务卸载到 RSU 或协作车辆,如果该 RSU 无法满足其要求,该方法会将该任务卸载到满足要求的 RSU。作者详细分析了系统总成本与计算资源价格和 RSU 带宽的演变,其中系统总成本随着价格和 RSU 带宽增加而增加,随后增长趋势变换。但是该方法优于贪心卸载与资源分配 (GORA) 和随机卸载与资源分配 (RORA)。文献[97]考虑了一个包含多辆车辆、多 MEC 服务和云服务器的场景,作者提出了一种多目标强化学习策略 (ICCRA),该策略包含 3 种方法。首先, K 近邻 (KNN) 方法选择车辆应用程序应在哪个层执行。如果选择卸载到其他车辆,则使用另一个非强化学习方法来选择车辆,同时考虑诸如车辆距离或速度等参数。最后,在选择将应用卸载到 MEC 层的情况下,使用了 Q-learning 方法来选择卸载应用的服务器或服务器组。从而降低系统的总成本。文献[98]考虑了一个基于移动边缘计算平台 3 层服务架构的任务卸载模型,作者提出一种基于 DQN 改进的 RSS2E-DQN 方法,该方法使用了同策略经验回放和熵正则,该方法能够对优秀的动作进行筛选,从而减少智能体与环境的交互。作者实验表明,随着设备应用程序的增加, RSS2E-DQN 方法相比 DQN、全部本地卸载和全部边缘服务器卸载,该方法卸载延迟、能耗和网络使用量最低。文献[99]考虑大量移动车辆将其任务卸载到边缘服务的场景,此场景下边缘服务器可能负载过高导致许多任务处理时间过长或者被丢弃,从而使总成本过高。作者提出一种基于变压器和策略解耦的多智能体 Actor-Critic (TPDMAAC) 的去中心化任务卸载方法,该方法能有效地将系统成本降低。文献[100]研究了一种包含多个具有计算资源的路侧单元 (边缘服务器) 和一个云服务器的移动边缘计算 (MEC) 系统,其中多个移动车辆可以卸载其应用程序进行执行。此外,除了车辆的移动性,作者还解决了应用程序迁移的问题。在此场景中,作者提出了一种基于 DQN 的方法,从而最小化系统的总成本。文献[101]提出一种适用于车载网络的 MEC 系统,其中考虑了多个 MEC 服务器,为最小化 MEC 框架的总成本,作者

提出了一种不需要先验知识的基于深度强化学习的计算迁移和资源分配 (RLCMRA) 方案。RLCMRA 通过自适应学习获得最优的卸载和迁移策略,使平均累计奖励最大化 (总成本最小化)。

文献[102]提出了多个车辆的场景。这些车辆可以将自身的任务卸载到边缘服务器或云服务器上。为了增强系统的安全性,作者使用区块链和智能合约方法进行访问控制。作者还提出了一种扩展深度 Q 网络 (EDRLCO) 方法,该方法结合了 Double DQN 和 Dueling DQN 的优势,以实现最优的卸载和资源分配决策,从而降低系统的延迟和能耗。文献[103]提出了多个边缘服务器和一个云服务器的车载网络场景,探讨了计算卸载和资源分配问题。作者提出一种分布式深度学习方法 (DQN) 寻找最优卸载决策,该方法收敛速度快,显著降低了整个系统的能量消耗。文献[104]提出了车载边缘计算 (VEC) 中的任务卸载和资源分配方法,开发了一种多资源编排框架,用来管理车辆、边缘服务器和云端的异构资源。针对车辆网络的动态性,作者提出异步优势 Actor-Critic (A3C) 方法寻找最优卸载策略。

文献[105]提出了一种基于多层深度强化学习 (DRL) 的云-边-端协同任务卸载方法 (MLDPBA),用于解决车载边缘网络 (VEN) 中动态环境下的联合任务卸载、调度与资源分配问题。传统方法难以应对在线异步决策和资源异构性的挑战,而该方法通过分层决策框架优化性能:第 1 层卸载模块采用双层双 Q 网络 (D3QN) 决定任务是否本地处理,或者卸载至边缘服务器、云服务器、其他车辆;第 2 层调度模块基于参数化深度 Q 网络 (PDQN) 选择 V2I/V2V 链路的传输目标与功率;第 3 层计算模块通过凸优化分配本地或协作车辆的计算资源。

文献[106]考虑多个 RSU 和一个云服务器的场景,为了在网络波动的情况下高效执行任务,作者提出 Stackelberg 博弈与强化学习结合的 (4-step-Stackelberg-MADDPG) 方法以促进车联网环境中本地、边缘和云资源之间的协同计算。该方法分为 4 阶段:全局状态感知、领导策略生成、跟随策略响应和联合奖励优化与策略更新。该方法相比 MADDPG、MADQN 和 MAA2C 在资源购买量上相对更高且稳定,任务卸载成功率最高,任务卸载延迟最低。文献[107]研究基于非正交多址接入 (NOMA) 的车联网 (VEC) 中的实时任务卸载和资源分配问题,对于任务卸载,作者将该问题建模为

边缘节点之间的非合作博弈,然后将边缘节点作为智能体,并提出一种多智能体分布式确定性策略梯度和博弈论融合的方法(MAD4PG),通过最大化基于势函数设计的奖励来确定任务卸载决策,从而实现纳什均衡.作者将资源分配问题分为传输功率分配和计算资源分配两个独立问题,对于传输功率分配,通过变换,将原问题转化为凸优化问题,利用拉格朗日对偶法和梯度下降法求解;对于计算资源分配,同样通过引入拉格朗日乘数,依据KKT条件得出最优解.文献[108]考虑了动态交通场景,其中MEC和CC协作处理车联网中的延迟敏感和计算密集任务.作者为了应对动态的环境和最小化系统的成本,提出了CORA方法.文

献[109]研究将车辆的部分计算任务卸载到边缘服务器或者云服务器上,还考虑车辆的移动速度,作者提出一种基于深度确定性策略梯度的任务卸载决策方法(TODM-DDPG),从而降低了系统成本.文献[110]提出了一种“端边云”协同的5G车联网边缘计算系统模型,还提出了D-SOAC方法,该方法考虑动态环境(网络波动、计算资源变化),还使用深度时空残差网络(ST-ResNet)预测未来服务需求,以适应这些变化来做出合理卸载决策.与LO、RO、ECO和BDR方法对比,该方法能够降低0.4%–20.4%的用户平均服务时延.基于云-边-端资源协同的主要算法的比较如表9所示.

表9 基于云-边-端资源协同计算卸载算法比较

探索方法	基础算法	优化目标	卸载模型	决策方式	车辆移动性	任务依赖性
文献[96]	Q-learning	最小化系统成本	完全卸载	集中决策	移动	独立
文献[97]	ICCRA	降低系统总成本	完全卸载	分布式决策	移动	独立
文献[98]	RSS2E-DQN	降低能耗和时延,提升服务质量	部分卸载	集中决策	移动	独立
文献[99]	TPDMAAC	最小化系统成本	部分卸载	分布式决策	移动	独立
文献[100]	DQN	最小化系统总成本	完全卸载	分布式决策	移动	独立
文献[101]	RLCMRA	最小化系统成本	完全卸载	集中决策	移动	独立
文献[102]	EDRLCO	最小化延迟、能耗和流量成本	完全卸载	集中决策	移动	独立
文献[103]	DQN	最小化系统的在时间和能耗的总成本	完全卸载	分布式决策	移动	独立
文献[104]	A3C	最小化系统成本	完全卸载	分布式决策	移动	独立
文献[105]	MLDPBA	降低系统成本	完全卸载	分布式决策	移动	独立
文献[106]	4-step-Stackelberg-MADDPG	在卸载延迟和成功率之间的平衡	部分卸载	分布式决策	移动	独立
文献[107]	MAD4PG	最大化系统服务率	部分卸载	分布式决策	移动	独立
文献[108]	CORA	最小化处理延迟和系统成本	部分卸载	集中决策	移动	独立
文献[109]	TODM-DDPG	最小化系统总成本	部分卸载	集中决策	移动	独立
文献[110]	D-SOAC	最小化服务延迟	部分卸载	分布式决策	移动	独立

4 未来与展望

车载边缘计算与任务卸载通过深度强化学习的动态决策能力,显著提升了任务处理效率.然而,在车载边缘计算的实际场景中还面临着许多挑战.

(1) 安全与隐私

由于车载边缘计算集中在决策过程中,需要采集服务请求车辆的隐私数据用于深度强化模型训练,这样就容易造成隐私数据的泄露.为了解决该问题,可以利用联邦学习技术.联邦学习是一种分布式深度学习,其特点在于终端设备可以用自己的数据在本地模型训练,并且所有终端设备能够同时上传和下载模型.然后,边缘服务器对每一轮训练中同步上传的模型参数执行模型参数聚合算法,可以聚合为一个全局最优模型,终端设备可以下载并基于新的共享模型进行新一轮训

练^[111].因此,终端设备在此过程中无需上传隐私数据,从而避免了隐私泄露的风险.

(2) 维度灾难

计算卸载环境的复杂性和动态特性使得深度强化学习车辆的状态-动作空间变得非常庞大,从而导致维度灾难,使得算法收敛缓慢.为了缓解这方面的维度灾难,可以采用分层强化学习将复杂的任务分解为多个子任务,每个子任务有更小的状态-动作空间,此方法可以有效减少维度灾难并加快算法收敛速度^[112].还可以采用近似分解和高效采样方法,近似分解方法可以将任何MDP分解为低维组件,因此可以将高维状态(任务计算量、服务器负载、网络延迟等)分解为低维的任务属性组件(计算需求、优先级)、服务器状态组件(当前负载、处理能力)、网络条件组件(带宽、

延迟)等.然后在单个样本里同时采取多个组件(例如同时从多个服务器上采取低维组件),从而将样本复杂度的依赖从每个组件大小总和减少到最大组件的大小^[113].该方法有效提高了强化学习的样本效率.

(3) 计算任务迁移

在边缘服务器高负载的情况下,为了平衡边缘服务器的负载,可以将需要迁移的任务从负载较高的服务器迁移到负载较低的服务器上,从而使服务器负载均衡.在高速移动的车辆环境中,当车辆任务在获取计算结果前离开现有边缘服务器的覆盖范围时,需要将任务迁移到车辆最近的边缘服务器^[114].网络功能虚拟化技术的提出为任务迁移奠定了基础.随着虚拟化的普及,出现了微服务的概念.微服务具有提高服务的模块化和弹性,增加服务管理的灵活性,并允许服务之间功能共享的特点.因此,微服务使得任务迁移变得更加灵活和易于管理.文献^[115]提出应用感知迁移和用户上下文感知迁移两种方法,应用感知迁移是先分析服务运行时数据,并将内存页分类为:对服务运行时非常重要的页面和没有这些内存页服务也能运行但性能会下降的页面等.然后对服务运行时非常重要的内存页会优先转移到其他服务器.用户上下文感知迁移是先确定用户上下文和内存页传输的优先级,一旦应用程序的核心部分已经迁移,这种方法将逐步把每个用户的后续应用程序迁移到目标点,从而减少因服务器负载触发的迁移和车辆高速移动导致的服务中断.

5 结束语

本文围绕基于深度强化学习的车载边缘计算与任务卸载技术展开系统综述.首先简要介绍了车联网架构、通信技术及车载边缘计算的核心概念,梳理了VEC的3层架构(远程云、边缘云、车辆云)及任务卸载的类型与决策方式.接着,阐述DRL的基础理论和方法分类,并探讨多智能体深度强化学习3种学习范式和智能体之间的竞争关系.基于上述理论框架,紧接着从车-车计算卸载、车-边缘层计算卸载和云-边缘资源协同计算卸载这3个方面,系统地梳理了国内外的研究进展.最后从3个方向梳理了该领域需要解决的挑战和未来研究方向.

参考文献

- 1 Wang B, Wang CH, Huang WW, *et al.* A survey and taxonomy on task offloading for edge-cloud computing. IEEE Access, 2020, 8: 186080–186101. [doi: 10.1109/ACCESS.2020.3029649]
- 2 Bojarski M, Del Testa D, Dworakowski D, *et al.* End to end learning for self-driving cars. arXiv:1604.07316, 2016.
- 3 Arena F, Collotta M, Pau G, *et al.* An overview of augmented reality. Computers, 2022, 11(2): 28. [doi: 10.3390/computers11020028]
- 4 Feng C, Han PC, Zhang X, *et al.* Computation offloading in mobile edge computing networks: A survey. Journal of Network and Computer Applications, 2022, 202: 103366. [doi: 10.1016/j.jnca.2022.103366]
- 5 Armbrust M, Fox A, Griffith R, *et al.* A view of cloud computing. Communications of the ACM, 2010, 53(4): 50–58. [doi: 10.1145/1721654.1721672]
- 6 李智勇, 王琦, 陈一凡, 等. 车辆边缘计算环境下任务卸载研究综述. 计算机学报, 2021, 44(5): 963–982. [doi: 10.11897/SP.J.1016.2021.00963]
- 7 Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628–1656. [doi: 10.1109/COMST.2017.2682318]
- 8 Ahn NY, Lee DH. Physical layer security of autonomous driving: Secure vehicle-to-vehicle communication in a security cluster. Ad-hoc and Sensor Wireless Networks, 2019, 45(3–4): 293–336.
- 9 Hortelano D, de Miguel I, Durán Barroso RJ, *et al.* A comprehensive survey on reinforcement-learning-based computation offloading techniques in edge computing systems. Journal of Network and Computer Applications, 2023, 216: 103669. [doi: 10.1016/j.jnca.2023.103669]
- 10 Li P, Wang X, Li C, *et al.* Deep reinforcement learning-based task scheduling and resource allocation for vehicular edge computing: A survey. IEEE Transactions on Intelligent Transportation Systems, 2025: 1–30. [doi: 10.1109/TITS.2025.3607910]
- 11 翟冠杰. 车联网体系结构分析及关键技术应用探讨. 电子测试, 2018, 29(23): 76–77. [doi: 10.3969/j.issn.1000-8519.2018.23.033]
- 12 Malhi AK, Batra S, Pannu HS. Security of vehicular ad-hoc networks: A comprehensive survey. Computers & Security, 2020, 89: 101664. [doi: 10.1016/j.cose.2019.101664]
- 13 Naik G, Choudhury B, Park JM. IEEE 802.11bd & 5G NR V2X: Evolution of radio access technologies for V2X communications. IEEE Access, 2019, 7: 70169–70184.
- 14 Barthe G, Dupressoir F, Grégoire B, *et al.* EasyCrypt: A

- tutorial. In: Aldini A, Lopez J, Martinelli F, eds. Foundations of Security Analysis and Design VII. Cham: Springer, 2014. 146–166.
- 15 陈山枝, 时岩, 胡金玲. 蜂窝车联网 (C-V2X) 综述. 中国科学基金, 2020, 34(2): 179–185.
 - 16 Nash JF Jr. Equilibrium points in n -person games. Proceedings of the National Academy of Sciences of the United States of America, 1950, 36(1): 48–49. [doi: [10.1073/pnas.36.1.48](https://doi.org/10.1073/pnas.36.1.48)]
 - 17 Samuelson L. Evolutionary Games and Equilibrium Selection. Cambridge: MIT Press, 1997.
 - 18 Gintis H. Game Theory Evolving: A Problem-centered Introduction to Modeling Strategic Behavior. Princeton: Princeton University Press, 2000.
 - 19 Talbi EG. Metaheuristics: From Design to Implementation. Hoboken: John Wiley & Sons, 2009.
 - 20 Bertsimas D, Tsitsiklis J. Simulated annealing. Statistical Science, 1993, 8(1): 10–15. [doi: [10.1214/ss/1177011077](https://doi.org/10.1214/ss/1177011077)]
 - 21 Reeves CR, Rowe JE. Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory. New York: Springer, 2002.
 - 22 Shi Y, Eberhart RC. Empirical study of particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation. Washington: IEEE, 1999. 1945–1950.
 - 23 Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Computational Intelligence Magazine, 2006, 1(4): 28–39. [doi: [10.1109/MCI.2006.329691](https://doi.org/10.1109/MCI.2006.329691)]
 - 24 Koszegi B. Behavioral contract theory. Journal of Economic Literature, 2014, 52(4): 1075–1118. [doi: [10.1257/jel.52.4.1075](https://doi.org/10.1257/jel.52.4.1075)]
 - 25 Zhang K, Mao YM, Leng SP, *et al.* Contract-theoretic approach for delay constrained offloading in vehicular edge computing networks. Mobile Networks and Applications, 2019, 24(3): 1003–1014. [doi: [10.1007/s11036-018-1032-0](https://doi.org/10.1007/s11036-018-1032-0)]
 - 26 Ladosz P, Weng LL, Kim M, *et al.* Exploration in deep reinforcement learning: A survey. Information Fusion, 2022, 85: 1–22. [doi: [10.1016/j.inffus.2022.03.003](https://doi.org/10.1016/j.inffus.2022.03.003)]
 - 27 Bellman R. The theory of dynamic programming. Bulletin of the American Mathematical Society, 1954, 60(6): 503–515.
 - 28 Sutton RS, Barto AG. Reinforcement Learning: An Introduction. 2nd ed., Cambridge: The MIT Press, 2018.
 - 29 Sutton RS. Learning to predict by the methods of temporal differences. Machine Learning, 1988, 3(1): 9–44.
 - 30 Watkins CJCH, Dayan P. Q-learning. Machine Learning, 1992, 8(3–4): 279–292. [doi: [10.1007/BF00992698](https://doi.org/10.1007/BF00992698)]
 - 31 van Hasselt HP. Insights in Reinforcement Learning: Formal Analysis and Empirical Evaluation of Temporal-difference Learning Algorithms. Utrecht: Universiteit Utrecht, 2011.
 - 32 Konda VR, Tsitsiklis JN. Actor-Critic algorithms. Proceedings of the 12th International Conference on Neural Information Processing Systems. The MIT Press, 1999. 1008–1014.
 - 33 梁星星, 冯旸赫, 马扬, 等. 多 Agent 深度强化学习综述. 自动化学报, 2020, 46(12): 2537–2557.
 - 34 Mnih V, Kavukcuoglu K, Silver D, *et al.* Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529–533. [doi: [10.1038/nature14236](https://doi.org/10.1038/nature14236)]
 - 35 van Hasselt H. Double Q-learning. Proceedings of the 24th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2010. 2613–2621.
 - 36 Wang ZY, Schaul T, Hessel M, *et al.* Dueling network architectures for deep reinforcement learning. Proceedings of the 33rd International Conference on International Conference on Machine Learning. New York: JMLR.org, 2016. 1995–2003.
 - 37 Hessel M, Modayil J, van Hasselt H, *et al.* Rainbow: Combining improvements in deep reinforcement learning. Proceedings of the 32nd AAAI Conference on Artificial Intelligence. New Orleans: AAAI Press, 2018. 393.
 - 38 Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 1992, 8(3–4): 229–256. [doi: [10.1007/BF00992696](https://doi.org/10.1007/BF00992696)]
 - 39 Schulman J, Wolski F, Dhariwal P, *et al.* Proximal policy optimization algorithms. arXiv:1707.06347, 2017.
 - 40 Schulman J, Levine S, Moritz P, *et al.* Trust region policy optimization. Proceedings of the 32nd International Conference on International Conference on Machine Learning. Lille: JMLR.org, 2015. 1889–1897.
 - 41 Shani L, Efroni Y, Mannor S. Adaptive trust region policy optimization: Global convergence and faster rates for regularized MDPs. Proceedings of the 2020 AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2020. 5668–5675. [doi: [10.1609/aaai.v34i04.6021](https://doi.org/10.1609/aaai.v34i04.6021)]
 - 42 Lillicrap TP, Hunt JJ, Pritzel A, *et al.* Continuous control with deep reinforcement learning. arXiv:1509.02971, 2015.
 - 43 Fujimoto S, Hoof H, Meger D. Addressing function approximation error in Actor-Critic methods. Proceedings

- of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018. 1587–1596.
- 44 Haarnoja T, Zhou A, Abbeel F, *et al.* Soft Actor-Critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. Proceedings of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018. 1861–1870.
- 45 Mnih V, Badia AP, Mirza M, *et al.* Asynchronous methods for deep reinforcement learning. Proceedings of the 33rd International Conference on International Conference on Machine Learning. New York: JMLR.org, 2016. 1928–1937.
- 46 Chen HQ, Liu YD, Zhou ZT, *et al.* A2C: Attention-augmented contrastive learning for state representation extraction. Applied Sciences, 2020, 10(17): 5902. [doi: [10.3390/app10175902](https://doi.org/10.3390/app10175902)]
- 47 Peng P, Lin WW, Wu WT, *et al.* A survey on computation offloading in edge systems: From the perspective of deep reinforcement learning approaches. Computer Science Review, 2024, 53: 100656.
- 48 Wang X, Wang S, Liang XX, *et al.* Deep reinforcement learning: A survey. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(4): 5064–5078. [doi: [10.1109/TNNLS.2022.3207346](https://doi.org/10.1109/TNNLS.2022.3207346)]
- 49 Li YX. Deep reinforcement learning: An overview. arXiv:1701.07274, 2017.
- 50 Lazaridis A, Fachantidis A, Vlahavas I. Deep reinforcement learning: A state-of-the-art walkthrough. Journal of Artificial Intelligence Research, 2020, 69: 1421–1471. [doi: [10.1613/jair.1.12412](https://doi.org/10.1613/jair.1.12412)]
- 51 Arulkumaran K, Deisenroth MP, Brundage M, *et al.* A brief survey of deep reinforcement learning. arXiv:1708.05866, 2017.
- 52 Luong NC, Hoang DT, Gong SM, *et al.* Applications of deep reinforcement learning in communications and networking: A survey. IEEE Communications Surveys & Tutorials, 2019, 21(4): 3133–3174. [doi: [10.1109/COMST.2019.2916583](https://doi.org/10.1109/COMST.2019.2916583)]
- 53 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述. 计算机学报, 2018, 41(1): 1–27. [doi: [10.11897/SP.J.1016.2019.00001](https://doi.org/10.11897/SP.J.1016.2019.00001)]
- 54 Dong H, Ding ZH, Zhang SH. Deep Reinforcement Learning: Fundamentals, Research and Applications. Singapore: Springer, 2020.
- 55 Szepesvári C. Algorithms for Reinforcement Learning. New York: Springer, 2022.
- 56 陈卓然, 刘泽阳, 万里鹏, 等. 多智能体强化学习理论及其应用综述. 模式识别与人工智能, 2024, 37(10): 851–872.
- 57 刘延飞, 李超, 王忠, 等. 多智能体深度强化学习及可扩展性研究进展. 计算机工程与应用, 2025, 61(4): 1–24. [doi: [10.3778/j.issn.1002-8331.2407-0034](https://doi.org/10.3778/j.issn.1002-8331.2407-0034)]
- 58 高玉钊, 聂一鸣. 基于值函数分解的多智能体深度强化学习方法研究综述. 计算机科学, 2024, 51(S1): 22–30. [doi: [10.11896/jsjx.230300170](https://doi.org/10.11896/jsjx.230300170)]
- 59 Mianji EM, Fardad M, Muntean GM, *et al.* A survey on multi-agent reinforcement learning applications in the internet of vehicles. Proceedings of the 99th IEEE Vehicular Technology Conference. Singapore: IEEE, 2024. 1–7. [doi: [10.1109/VTC2024-Spring62846.2024.10683524](https://doi.org/10.1109/VTC2024-Spring62846.2024.10683524)]
- 60 Gronauer S, Diepold K. Multi-agent deep reinforcement learning: A survey. Artificial Intelligence Review, 2022, 55(2): 895–943. [doi: [10.1007/s10462-021-09996-w](https://doi.org/10.1007/s10462-021-09996-w)]
- 61 Oroojlooy A, Hajinezhad D. A review of cooperative multi-agent deep reinforcement learning. Applied Intelligence, 2023, 53(11): 13677–13722. [doi: [10.1007/s10489-022-04105-y](https://doi.org/10.1007/s10489-022-04105-y)]
- 62 熊丽琴, 曹雷, 赖俊, 等. 基于值分解的多智能体深度强化学习综述. 计算机科学, 2022, 49(9): 172–182. [doi: [10.11896/jsjx.210800112](https://doi.org/10.11896/jsjx.210800112)]
- 63 Littman ML. Markov games as a framework for multi-agent reinforcement learning. Proceedings of the 11th International Conference on International Conference on Machine Learning. New Brunswick: Morgan Kaufmann, 1994. 157–163.
- 64 Du W, Ding SF. A survey on multi-agent deep reinforcement learning: From the perspective of challenges and applications. Artificial Intelligence Review, 2021, 54(5): 3215–3238. [doi: [10.1007/s10462-020-09938-y](https://doi.org/10.1007/s10462-020-09938-y)]
- 65 杨一桐, 吴菁晶, 张立立. 基于 V2X 的车联网实验系统设计与实现. 现代电子技术, 2024, 47(24): 33–37. [doi: [10.16652/j.issn.1004-373x.2024.24.006](https://doi.org/10.16652/j.issn.1004-373x.2024.24.006)]
- 66 Mao YY, You CS, Zhang J, *et al.* A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322–2358. [doi: [10.1109/COMST.2017.2745201](https://doi.org/10.1109/COMST.2017.2745201)]
- 67 Tang D, Zhang XF, Li M, *et al.* Adaptive inference reinforcement learning for task offloading in vehicular edge computing systems. Proceedings of the 2020 IEEE International Conference on Communications Workshops. Dublin: IEEE, 2020. 1–6. [doi: [10.1109/ICCWorkshops49005.2020.9145133](https://doi.org/10.1109/ICCWorkshops49005.2020.9145133)]
- 68 师亚莉, 肖春艳, 王宇, 等. 车联网中基于深度强化学习的计算卸载与资源分配算法. 西安邮电大学学报, 2024, 29(4): 30–38.

- 69 Chen C, Zhong YR, Wang Z, *et al.* Distributed computation offloading method based on deep reinforcement learning in ICV. *Applied Soft Computing*, 2021, 103: 107108. [doi: [10.1016/j.asoc.2021.107108](https://doi.org/10.1016/j.asoc.2021.107108)]
- 70 Hou YK, Wei ZW, Zhang RQ, *et al.* Hierarchical task offloading for vehicular fog computing based on multi-agent deep reinforcement learning. *IEEE Transactions on Wireless Communications*, 2024, 23(4): 3074–3085. [doi: [10.1109/TWC.2023.3305321](https://doi.org/10.1109/TWC.2023.3305321)]
- 71 Dai F, Liu GZ, Mo Q, *et al.* Task offloading for vehicular edge computing with edge-cloud cooperation. *World Wide Web*, 2022, 25(5): 1999–2017. [doi: [10.1007/s11280-022-01011-8](https://doi.org/10.1007/s11280-022-01011-8)]
- 72 Liu WR, Shao XJ, Wang CL, *et al.* An online reinforcement learning offloading method for delay-sensitive vehicular service. *Proceedings of the 22nd IEEE International Conference on High Performance Computing and Communications; the 18th IEEE International Conference on Smart City; the 6th IEEE International Conference on Data Science and Systems*. Yanuca Island: IEEE, 2020. 973–978. [doi: [10.1109/HPCC-SmartCity-DSS50907.2020.00130](https://doi.org/10.1109/HPCC-SmartCity-DSS50907.2020.00130)]
- 73 Zhang XF, Lin WL, Li YJ, *et al.* Moving server: Follow-up computation offloading paradigm for vehicular users. *Proceedings of the 2020 IEEE/CIC International Conference on Communications in China*. Chongqing: IEEE, 2020. 226–231. [doi: [10.1109/ICCC49849.2020.9238895](https://doi.org/10.1109/ICCC49849.2020.9238895)]
- 74 Zhan WH, Luo CB, Wang J, *et al.* Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing. *IEEE Internet of Things Journal*, 2020, 7(6): 5449–5465. [doi: [10.1109/JIOT.2020.2978830](https://doi.org/10.1109/JIOT.2020.2978830)]
- 75 Liu HQ, Zhao HB, Geng LW, *et al.* A policy gradient based offloading scheme with dependency guarantees for vehicular networks. *Proceedings of the 2020 IEEE Globecom Workshops*. Taipei: IEEE, 2020. 1–6. [doi: [10.1109/GCWkshps50303.2020.9367544](https://doi.org/10.1109/GCWkshps50303.2020.9367544)]
- 76 Liu HQ, Zhao HB, Geng LW, *et al.* A distributed dependency-aware offloading scheme for vehicular edge computing based on policy gradient. *Proceedings of the 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/the 7th IEEE International Conference on Edge Computing and Scalable Cloud*. Washington: IEEE, 2021. 176–181. [doi: [10.1109/CSCloud-EdgeCom52276.2021.00040](https://doi.org/10.1109/CSCloud-EdgeCom52276.2021.00040)]
- 77 Zhu XY, Luo YY, Liu AF, *et al.* Multiagent deep reinforcement learning for vehicular computation offloading in IoT. *IEEE Internet of Things Journal*, 2021, 8(12): 9763–9773. [doi: [10.1109/JIOT.2020.3040768](https://doi.org/10.1109/JIOT.2020.3040768)]
- 78 琚翔, 苏圣超, 徐超杰, 等. 边缘计算下基于深度强化学习的车联网任务调度. *系统仿真学报*, 2023, 35(12): 2550–2559.
- 79 郭晓东, 郝思达, 王丽芳. 基于深度强化学习的车辆边缘计算任务卸载方法. *计算机应用研究*, 2023, 40(9): 2803–2807, 2814.
- 80 Kazmi SMA, Otoum S, Hussain R, *et al.* A novel deep reinforcement learning-based approach for task-offloading in vehicular networks. *Proceedings of the 2021 IEEE Global Communications Conference*. Madrid: IEEE, 2021. 1–6.
- 81 Wang BX, Liu L, Wang J. Actor-critic based DRL algorithm for task offloading performance optimization in vehicle edge computing. *Proceedings of the 2023 International Wireless Communications and Mobile Computing*. Marrakesh: IEEE, 2023. 93–98. [doi: [10.1109/IWCMC58020.2023.10183228](https://doi.org/10.1109/IWCMC58020.2023.10183228)]
- 82 Uddin A, Sakr AH, Zhang N. Adaptive prioritization and task offloading in vehicular edge computing through deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 2025, 74(3): 5038–5052. [doi: [10.1109/TVT.2024.3499962](https://doi.org/10.1109/TVT.2024.3499962)]
- 83 Wang SY, Yang B, Yu ZW, *et al.* Computation offloading for multi-server multi-access edge vehicular networks: A DDQN-based method. *Proceedings of the 99th IEEE Vehicular Technology Conference*. Singapore: IEEE, 2024. 1–5. [doi: [10.1109/VTC2024-Spring62846.2024.10683288](https://doi.org/10.1109/VTC2024-Spring62846.2024.10683288)]
- 84 Huang XY, He LJ, Zhang WY. Vehicle speed aware computing task offloading and resource allocation based on multi-agent reinforcement learning in a vehicular edge computing network. *Proceedings of the 2020 IEEE International Conference on Edge Computing*. Beijing: IEEE, 2020. 1–8. [doi: [10.1109/EDGE50951.2020.00008](https://doi.org/10.1109/EDGE50951.2020.00008)]
- 85 Wang JF, Lv TJ, Huang PM, *et al.* Mobility-aware partial computation offloading in vehicular networks: A deep reinforcement learning based scheme. *China Communications*, 2020, 17(10): 31–49. [doi: [10.23919/JCC.2020.10.003](https://doi.org/10.23919/JCC.2020.10.003)]
- 86 De Souza AB, Rego PAL, Carneiro T, *et al.* Computation offloading for vehicular environments: A survey. *IEEE Access*, 2020, 8: 198214–198243. [doi: [10.1109/ACCESS.2020.3033828](https://doi.org/10.1109/ACCESS.2020.3033828)]
- 87 Fan WH, Zhang YY, Zhou GT, *et al.* Deep reinforcement learning-based task offloading for vehicular edge computing with flexible RSU-RSU cooperation. *IEEE Transactions on*

- Intelligent Transportation Systems, 2024, 25(7): 7712–7725. [doi: [10.1109/TITS.2024.3349546](https://doi.org/10.1109/TITS.2024.3349546)]
- 88 Geng LW, Zhao HB, Liu HQ, *et al.* Deep reinforcement learning-based computation offloading in vehicular networks. Proceedings of the 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/the 7th IEEE International Conference on Edge Computing and Scalable Cloud. Washington: IEEE, 2021. 200–206. [doi: [10.1109/CSCloud-EdgeCom52276.2021.00044](https://doi.org/10.1109/CSCloud-EdgeCom52276.2021.00044)]
- 89 Ke HC, Wang J, Deng LY, *et al.* Deep reinforcement learning-based adaptive computation offloading for MEC in heterogeneous vehicular networks. IEEE Transactions on Vehicular Technology, 2020, 69(7): 7916–7929. [doi: [10.1109/TVT.2020.2993849](https://doi.org/10.1109/TVT.2020.2993849)]
- 90 Wu ZP, Jia ZP, Pang XY, *et al.* Deep reinforcement learning-based task offloading and load balancing for vehicular edge computing. Electronics, 2024, 13(8): 1511. [doi: [10.3390/electronics13081511](https://doi.org/10.3390/electronics13081511)]
- 91 张依琳, 梁玉珠, 尹沐君, 等. 移动边缘计算中计算卸载方案研究综述. 计算机学报, 2021, 44(12): 2406–2430. [doi: [10.11897/SP.J.1016.2021.02406](https://doi.org/10.11897/SP.J.1016.2021.02406)]
- 92 Shi MY, Wang R, Liu EW, *et al.* Deep reinforcement learning based computation offloading for mobility-aware edge computing. Proceedings of the 14th EAI International Conference on Communications and Networking. Shanghai: Springer, 2019. 53–65.
- 93 Peng HX, Shen XM. Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks. IEEE Transactions on Network Science and Engineering, 2020, 7(4): 2416–2428. [doi: [10.1109/TNSE.2020.2978856](https://doi.org/10.1109/TNSE.2020.2978856)]
- 94 Li MS, Gao J, Zhao L, *et al.* Deep reinforcement learning for collaborative edge computing in vehicular networks. IEEE Transactions on Cognitive Communications and Networking, 2020, 6(4): 1122–1135. [doi: [10.1109/TCCN.2020.3003036](https://doi.org/10.1109/TCCN.2020.3003036)]
- 95 Wang K, Wang XF, Liu X, *et al.* Task offloading strategy based on reinforcement learning computing in edge computing architecture of internet of vehicles. IEEE Access, 2020, 8: 173779–173789. [doi: [10.1109/ACCESS.2020.3023939](https://doi.org/10.1109/ACCESS.2020.3023939)]
- 96 Li YR, Xu SZ. Collaborative optimization of edge-cloud computation offloading in Internet of Vehicle. Proceedings of the 2021 International Conference on Computer Communications and Networks. Athens: IEEE, 2021. 1–6.
- 97 Cui YP, Du LJ, Wang HG, *et al.* Reinforcement learning for joint optimization of communication and computation in vehicular networks. IEEE Transactions on Vehicular Technology, 2021, 70(12): 13062–13072. [doi: [10.1109/TVT.2021.3125109](https://doi.org/10.1109/TVT.2021.3125109)]
- 98 蒋守花, 舒晖. 基于深度强化学习的改进移动边缘计算任务卸载算法研究. 软件导刊, 2024, 23(9): 150–156. [doi: [10.11907/rjdk.232294](https://doi.org/10.11907/rjdk.232294)]
- 99 Gao Z, Yang L, Dai Y. Fast adaptive task offloading and resource allocation via multiagent reinforcement learning in heterogeneous vehicular fog computing. IEEE Internet of Things Journal, 2023, 10(8): 6818–6835. [doi: [10.1109/JIOT.2022.3228246](https://doi.org/10.1109/JIOT.2022.3228246)]
- 100 Wu ZY, Yan DF. Deep reinforcement learning-based computation offloading for 5G vehicle-aware multi-access edge computing network. China Communications, 2021, 18(11): 26–41. [doi: [10.23919/JCC.2021.11.003](https://doi.org/10.23919/JCC.2021.11.003)]
- 101 Wang H, Ke HC, Liu G, *et al.* Computation migration and resource allocation in heterogeneous vehicular networks: A deep reinforcement learning approach. IEEE Access, 2020, 8: 171140–171153. [doi: [10.1109/ACCESS.2020.3024683](https://doi.org/10.1109/ACCESS.2020.3024683)]
- 102 Zheng X, Li MC, Chen YF, *et al.* Blockchain-based secure computation offloading in vehicular networks. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(7): 4073–4087. [doi: [10.1109/TITS.2020.3014229](https://doi.org/10.1109/TITS.2020.3014229)]
- 103 Khayyat M, Elgendy IA, Muthanna A, *et al.* Advanced deep learning-based computational offloading for multilevel vehicular edge-cloud computing networks. IEEE Access, 2020, 8: 137052–137062. [doi: [10.1109/ACCESS.2020.3011705](https://doi.org/10.1109/ACCESS.2020.3011705)]
- 104 Liu L, Feng J, Mu XY, *et al.* Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(12): 15513–15526. [doi: [10.1109/TITS.2023.3249745](https://doi.org/10.1109/TITS.2023.3249745)]
- 105 Wu JQ, Tang M, Jiang CK, *et al.* Cloud-edge-end collaborative task offloading in vehicular edge networks: A multilayer deep reinforcement learning approach. IEEE Internet of Things Journal, 2024, 11(22): 36272–36290. [doi: [10.1109/JIOT.2024.3472472](https://doi.org/10.1109/JIOT.2024.3472472)]
- 106 Fofana N, Ben Letaifa A, Rachedi A. Intelligent task offloading in vehicular networks: A deep reinforcement learning perspective. IEEE Transactions on Vehicular Technology, 2025, 74(1): 201–216. [doi: [10.1109/TVT.2024.3427814](https://doi.org/10.1109/TVT.2024.3427814)]

- 107 Xu XC, Liu K, Dai PL, *et al.* Joint task offloading and resource optimization in NOMA-based vehicular edge computing: A game-theoretic DRL approach. *Journal of Systems Architecture*, 2023, 134: 102780. [doi: [10.1016/j.sysarc.2022.102780](https://doi.org/10.1016/j.sysarc.2022.102780)]
- 108 Huang JW, Wan JY, Lv BF, *et al.* Joint computation offloading and resource allocation for edge-cloud collaboration in Internet of vehicles via deep reinforcement learning. *IEEE Systems Journal*, 2023, 17(2): 2500–2511. [doi: [10.1109/JSYST.2023.3249217](https://doi.org/10.1109/JSYST.2023.3249217)]
- 109 Shi W, Chen L, Zhu X. Task offloading decision-making algorithm for vehicular edge computing: A deep-reinforcement-learning-based approach. *Sensors*, 2023, 23(17): 7595. [doi: [10.3390/s23177595](https://doi.org/10.3390/s23177595)]
- 110 许小龙, 方子介, 齐连永, 等. 车联网边缘计算环境下基于深度强化学习的分布式服务卸载方法. *计算机学报*, 2021, 44(12): 2382–2405. [doi: [10.11897/SP.J.1016.2021.02382](https://doi.org/10.11897/SP.J.1016.2021.02382)]
- 111 Lin ZX, Cui HX, Liu Y. Distributed deep learning based on edge computing over Internet of vehicles: Overview, applications, and challenges. *IEEE Access*, 2024, 12: 133734–133747. [doi: [10.1109/ACCESS.2024.3454790](https://doi.org/10.1109/ACCESS.2024.3454790)]
- 112 He Y, Wang YH, Lin QZ, *et al.* Meta-hierarchical reinforcement learning (MHRL)-based dynamic resource allocation for dynamic vehicular networks. *IEEE Transactions on Vehicular Technology*, 2022, 71(4): 3495–3506. [doi: [10.1109/TVT.2022.3146439](https://doi.org/10.1109/TVT.2022.3146439)]
- 113 Lu CB, Shi LX, Chen ZW, *et al.* Overcoming the curse of dimensionality in reinforcement learning through approximate factorization. *Proceedings of the 42nd International Conference on Machine Learning*. Vancouver: OpenReview.net, 2024. [doi: [10.48550/arXiv.2411.07591](https://doi.org/10.48550/arXiv.2411.07591)]
- 114 Ahmed M, Raza S, Mirza MA, *et al.* A survey on vehicular task offloading: Classification, issues, and challenges. *Journal of King Saud University—Computer and Information Sciences*, 2022, 34(7): 4135–4162. [doi: [10.1016/j.jksuci.2022.05.016](https://doi.org/10.1016/j.jksuci.2022.05.016)]
- 115 Toumi N, Bagaa M, Ksentini A. Machine learning for service migration: A survey. *IEEE Communications Surveys & Tutorials*, 2023, 25(3): 1991–2020. [doi: [10.1109/COMST.2023.3273121](https://doi.org/10.1109/COMST.2023.3273121)]

(校对责编: 王欣欣)