

文章编号:1671-1637(2026)06-0153-14

基于云边协同的远程驾驶接管任务 自适应卸载模型

赵红专^{*1,2}, 汪懿晨^{1,2}, 张继康^{1,2}, 袁 泉³, 王建强³, 杨良义⁴,
王 涛^{1,2}, 周 旦^{1,2}

(1. 桂林电子科技大学 广西智慧交通重点实验室, 广西 桂林 541004; 2. 桂林电子科技大学
广西交通运输行业全息感知与智能驾驶重点实验室, 广西 桂林 541004; 3. 清华大学
车辆与运载学院, 北京 100084; 4. 中国汽车工程研究院股份有限公司, 重庆 401122)

摘 要:建立了基于云边协同的自适应任务卸载模型,分析了远程驾驶接管过程中因网络波动与算力不足引起的高时延与连接不稳定问题;定义了实时控制类、计算密集型与交互服务类3类任务,并设定了紧急与一般两级优先级体系,以精确区分不同任务对时延和可靠性的差异化需求;构建了融合云端中心、边缘节点与车载终端的协同计算环境,提出了基于任务优先级动态分配计算节点、结合实时网络带宽与边缘负载自适应调整资源权重的分层卸载规则,并研究了基于备用节点的断点续传机制,以增强系统在不稳定环境下的鲁棒性;采用动态规划算法构建了以全局最小时延为优化目标的决策模型,设定了相应的奖励函数以量化评估不同卸载策略的有效性;基于任务数据量、处理器频率等11项参数构建了专门的数据集,并设计了对比试验,系统研究了模型在动态负载与不同资源状态下的性能表现。研究表明:在边缘负载动态变化的场景下,所提自适应卸载策略获得的奖励值相较于传统固定阈值边缘计算方法提升13.2%;在引入云端协同计算后,系统整体奖励值相较于仅使用边缘计算的方案提升23.6%;特别在边缘节点负载超过60%时,该策略能够有效降低任务阻塞率达45%。

关键词:智能交通;云边协同计算;动态规划算法;远程驾驶接管;网联自动驾驶汽车

中图分类号:U491.8 **文献标志码:**A **DOI:**10.19818/j.cnki.1671-1637.2026.030

Adaptive offloading model for remote driving takeover task based on cloud-edge collaboration

ZHAO Hong-zhuan^{*1,2}, WANG Yi-chen^{1,2}, ZHANG Ji-kang^{1,2}, YUAN Quan³,
WANG Jian-qiang³, YANG Liang-yi⁴, WANG Tao^{1,2}, ZHOU Dan^{1,2}

(1. Guangxi Key Laboratory of ITS, Guilin University of Electronic Technology, Guilin 541004, Guangxi, China; 2. Key Laboratory of Holographic Perception and Intelligent Driving, Transportation Department of Guangxi Zhuang Autonomous Region, Guilin University of Electronic Technology, Guilin 541004, Guangxi, China; 3. School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China; 4. China Automotive Engineering Research Institute Co., Ltd., Chongqing 401122, China)

出版历程:2025-03-19 收稿, 2025-07-01 修回, 2025-08-25 录用

基金项目:国家自然科学基金项目(52362045);广西科技重大专项(桂科 AA23062053, 桂科 AA22068101);广西重点研发计划(桂科 AB25069283);北京市自然科学基金项目(L247007);广西精密导航技术与应用重点实验室开放课题(DH202225)

*** 作者简介:**赵红专(1985-),男,广西桂林人,教授,工学博士,博士后, E-mail: zhaohongzhuan@guet.edu.cn.

引用格式:赵红专,汪懿晨,张继康,等.基于云边协同的远程驾驶接管任务自适应卸载模型[J].交通运输工程学报,2026,26(6):153-166.

Citation: ZHAO Hong-zhuan, WANG Yi-chen, ZHANG Ji-kang, et al. Adaptive offloading model for remote driving takeover task based on cloud-edge collaboration[J]. Journal of Traffic and Transportation Engineering, 2026, 26(6): 153-166.

Abstract: An adaptive task offloading model based on cloud-edge collaboration was established, and the problems of high latency and unstable connection during remote driving takeover caused by network fluctuation and insufficient computing power were deeply analyzed. Three types of tasks, namely real-time control, computation-intensive, and interactive service tasks, were defined, and a two-level priority system of urgent and general levels was set to accurately distinguish the differentiated requirements of different tasks for latency and reliability. A collaborative computing environment integrating cloud center, edge node, and onboard terminal was constructed; a hierarchical offloading rule dynamically allocating computing nodes based on task priority and adaptively adjusting resource weights combined with real-time network bandwidth and edge load was proposed; a breakpoint resume mechanism based on backup nodes was studied to enhance the robustness of the system in unstable environments. A decision model with global minimum latency as the optimization objective was constructed using a dynamic programming algorithm, and a corresponding reward function was set to quantitatively evaluate the effectiveness of different offloading strategies. A dedicated dataset was constructed based on 11 parameters such as task data volume and processor frequency, and comparative experiments were designed to systematically study the performance of the model under dynamic load and different resource states. The research results indicate that under the scenario of dynamically changing edge load, the reward value obtained by the proposed adaptive offloading strategy is increased by 13.2% compared to the traditional fixed-threshold edge computing method; after introducing cloud collaborative computing, the overall reward value of the system is increased by 23.6% compared to the edge-only computing scheme; especially when the edge node load exceeds 60%, the proposed strategy can effectively reduce the task blocking rate by 45%.

Keywords: intelligent transportation; cloud-edge collaborative computing; dynamic programming algorithm; remote driving takeover; connected and automated vehicle

Publication history: Received 2025-03-19; Received in revised form 2025-07-01; Accepted 2025-08-25

Funding: National Natural Science Foundation of China (52362045); Science and Technology Major Project of Guangxi Province (Guike AA23062053, Guike AA22068101); Key Research and Development Program of Guangxi Province (Guike AB25069283); Natural Science Foundation of Beijing (L247007); Open Project of Guangxi Key Laboratory of Precision Navigation Technology and Application (DH202225)

* **Corresponding author:** ZHAO Hong-zhuan, professor, PhD, E-mail: zhaohongzhuan@guet.edu.cn.

0 引 言

近年来,在自动驾驶技术与车联网技术快速发展的双重驱动下,驾驶安全的重要性在现代交通领域中被提升至新的高度。在此背景下,通过技术创新强化交通安全保障机制已成为行业发展的核心命题,其应用价值与实践意义愈发凸显^[1-3]。远程驾驶接管技术是一种新兴的自动驾驶技术补充及安全保障手段,能够在复杂路况或系统故障时确保车辆的安全运行。随着车联网环境下的网联自动驾驶车辆(Connected and Automated Vehicle, CAV)风险预判^[4-5]等方法的进步,计算密集型与时延敏感型任务

的数量和比重不断增加^[6-7],当网联自动驾驶车辆算力不足以支撑所有计算任务时,需要将计算任务卸载到路侧边缘设备进行计算。现有许多专家学者也进行了深入的研究,但由于车辆快速行驶导致网络拓扑变化较快,仅仅将任务卸载的目标着眼于移动边缘端,数据时效性和准确性还是难以保证,此外值得考虑的是,边缘端负载状态也是影响其计算效率的重要影响因素,在短时间内接收大量任务请求易引发过载导致任务等待时间增加或任务失败^[8-10],从而对远程驾驶接管过程产生不利影响^[11-12],同时也不利于网联自动驾驶车辆通过技术手段规避或降低驾驶风险^[13]。目前,由于通信技术的快速发展,

在现有研究的基础上,结合云端强大的算力进行云边协同计算,是现在非常有前景的解决方案,也是现有专家学者的主要研究方向。本文所研究的远程驾驶接管自动驾驶技术,需要对上述时延敏感型任务实时进行反馈,因此基于远程驾驶接管场景,现有针对云边协同计算的研究,还存在一些亟待解决的挑战问题。

对车联网环境下的计算密集型和时延敏感型任务而言,实时处理、分析和收集数据过程中的任何延迟都是无法容忍的,但是由于车载终端的限制,无法满足新型车载应用对计算资源和时延等方面的需求,因此将任务卸载到边缘端成了研究重点。2017年欧洲电信标准协会 ETSI 提出的移动边缘计算(Mobile Edge Computing, MEC)是一种基于 5G 演进的架构^[14],作为一种新兴的计算范式可以为用户提供类似于云计算环境的大量存储和计算资源。陈玉平等^[15]概括了云边协同中各层次包含的6种协同技术,分别为:资源协同、数据协同、智能协同、业务编排协同、应用管理协同以及服务协同,其中计算卸载属于资源协同。目前车联网与 MEC 结合使用的相关研究引起学术界与工业界的高度重视。MEC 的资源部署在网络的边缘,即物理上更接近用户终端。在满足任务的实时要求的情况下,也减轻了大量终端设备给核心网带来的负荷^[16],同时数据处理的能耗和时延问题再次受到越来越多的关注。针对自动驾驶领域中面临的计算能力不足等挑战,Dai 等^[17]提出一种以 MEC 为框架的任务调度算法,该算法结合了自动驾驶任务的特性,并利用改进的最早截止时间优先算法实现了更有效的任务分配和执行。

复杂的通信场景下往往伴随着高能耗及高负载问题,优化任务卸载策略旨在降低移动边缘计算系统执行任务的能耗,降低系统内总时延,及更合理高效地分配系统计算资源^[18-21]。然而,现有研究对边缘端服务器负载的建模多基于静态阈值或理想化假设,缺乏对实际动态负载场景的适配能力。将算力固化的有 Xu 等^[22]提出的博弈论方法,虽实现云-边-端协同卸载,但其边缘算力被设定为固定参数,未考虑多车接入或突发任务(如密集障碍物识别)导致的算力瞬时波动,试验表明,当边缘负载从 30% 骤增至 90% 时,其任务卸载时延增加约 45%。对边缘端负载预测与弹性伸缩不足也是现有研究存在的问题,如唐宁昆等^[23]的多智能体策略依赖阈值触发机制(如负载大于 80% 时拒绝任务),但未引入时序

预测模型预判负载趋势,导致资源预留策略僵化。Zhao 等^[24]的云边协调方案虽支持云端接管,但缺乏断点续传机制,任务迁移需重新初始化,造成额外 15%~20% 的时延开销。此外,Lai 等^[25]的安全卸载框架在边缘过载时仅简单切换至云端,未设计多节点协同的任务分片与并行计算策略,导致资源利用率不足。

尽管移动边缘计算产生的延迟比云计算更小,但可以提供的算力及计算资源远不如云计算。协同边缘计算和云计算可有效提升整体计算效率降低系统延迟,但现有研究对动态网络条件的适配多停留在理想化假设层面,难以应对实际车联网环境的复杂性^[26],车联网中的网络拓扑动态变化较快,固定的带宽和网络静态拓扑假设显然不符合对该环境的考虑。此外,现有研究存在多模态通信适配缺失和跨层联合优化不足等问题,例如 Lai 等^[25]的安全卸载框架虽集成通信优化,但未设计异构网络(如 5G/C-V2X)的快速切换机制。车联网中的用户将任务需求卸载到云计算中心,云计算中心将处理后的结果传回用户端。但是,云计算中心与用户之间的距离较长,多个用户同时向云计算中心发出业务请求会造成较大的延迟,使用户无法及时收到云计算中心的结果,因此,将移动设备生成的所有任务卸载到云端是不明智的。为了弥补云计算的不足,Zhao 等^[24]通过联合优化任务卸载决策和资源优化问题来解决车载网络中的服务请求问题,提出云边协调计算卸载方案,基于该方案设计分布式优化方法,最终得到最优解。Xu 等^[22]研究了云端协作场景下多个移动设备的部分卸载问题,通过博弈论,首先得到边缘-终端的最优卸载决策,然后找到所有选择卸载到边缘的终端设备的云-边缘最优卸载决策,最后达到云-边缘-终端三方面的最优卸载目标。

由以上研究可知,众多学者从边缘计算到移动边缘计算再到云边协同计算验证了将网络卸载到边缘端以及云端对时延优化的可行性,但基于远程驾驶接管场景,现有针对云边协同计算的研究还存在3个主要问题。第一是现有研究缺乏对任务优先级的考虑,即一些时延敏感型任务需要更高的计算优先级,不确定或不稳定的任务优先级次序易导致远程驾驶接管过程出现任务冲突、进程失效等驾驶安全下降问题;第二是现有研究较少考虑边缘端任务动态负载情况,普遍默认边缘端算力为固定,然而在实际场景下边缘端会有其他任务占用计算进程的情况,使边缘算力的大小出现浮动,同样也会影响到本

文研究的计算任务卸载;第三是现有研究通信场景较为固定,难以实现任务根据不同通信环境进行自适应卸载,以确保远程接管过程的稳定运行。为了提升远程驾驶接管任务的可靠性,需深入研究一种灵活且全面的任务处理方案来捕捉任务优先级关系对于时延性能的影响。鉴于此,提出并建立了一种基于云边协同的远程驾驶接管的自适应卸载模型。本文开展的主要研究工作如下:

(1)针对先前研究较少关注的驾驶接管任务优先级的影响因素,将多维度任务分类与优先级动态绑定,对远程驾驶接管过程中的任务类型分类并与优先级绑定,确保了驾驶接管任务的优先性。

(2)针对先前研究中较少考虑边缘计算动态负载问题,设计并建立一种远程驾驶接管的任务自适应卸载模型,利用动态规划算法设置奖励函数,获得全局时延最低的最优任务卸载策略。

(3)针对先前研究中研究场景通信条件单一的问题,搭建了一种考虑远程驾驶接管技术的云边协同计算下的车路协同驾驶场景,经仿真验证云边协同计算相较于单一边缘计算可为远程驾驶接管提供了更低的通信时延。

1 云边协同计算系统模型设计

远程驾驶接管自动驾驶过程中,需确保时延敏感型任务(如方向盘控制、油门/制动信号等)在极低时延内完成实时反馈,这对系统通信与计算资源提出严苛要求。为此,本节构建了云边协同计算环境,综合考虑车端、边缘端及云端算力差异,结合任务优先级动态规划机制,以全局时延最小化为目标优化任务卸载策略。通过动态规划算法,系统能够灵活分配任务至本地、边缘或云端节点,显著降低时延并提升接管可靠性。

1.1 云边协同计算系统构建

相较于现阶段主流研究的车联网下云边协同计算架构,本研究将远程驾驶端加入传统车联网交通环境中,在该环境下可发挥 5G 网络的高带宽、低延迟特性提升远程驾驶接管指令的精确性,考虑负载的边缘端服务器和云端服务器可实现复杂交通环境下远程驾驶接管任务优先的目的和保证任务进程顺利进行。

针对云边计算的特殊场景,首先考虑远程驾驶接管过程的体系结构,构建云边协同计算环境,同时根据任务类型、任务优先级定义任务卸载规则,为实现任务自适应卸载提供模型基础。

云边协同计算环境中所包含的完整的体系结构,由云端服务器、边缘节点^[18]、车载单元(On-board Unit, OBU)、智能网联车辆计算任务、远程驾驶端和网络通信技术等组成,其中边缘节点由路侧单元(Road Side Unit, RSU)和边缘计算服务器组成。网络通信技术以 C-V2X(Cellular Vehicle-to-everything)接口为标准,进行车车通信以及车路通信和车路云通信,通过这些计算单元、通信单元、存储单元的协同作用,可以实现高效、可靠的云边协同计算任务,云边协同计算车路云通信概念图,如图 1 所示。本架构在 ETSI-MEC 标准框架^[14]基础上扩展,新增远程驾驶端交互模块及备用 RSU 节点。

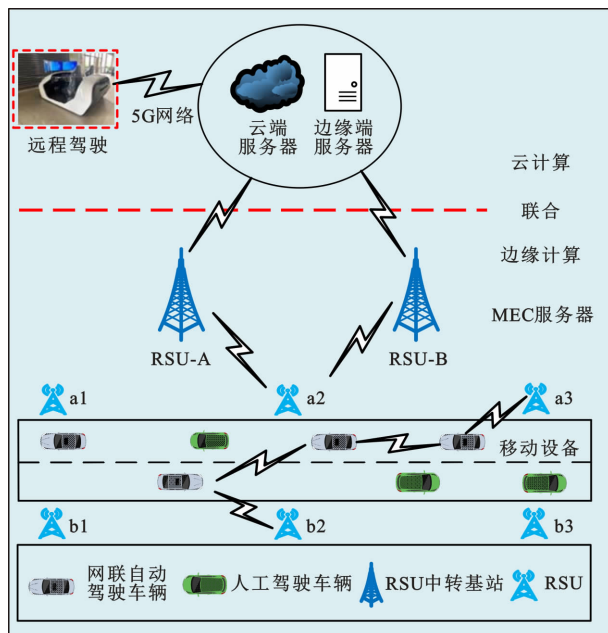


图 1 云边协同计算车路云通信示意

Fig. 1 Schematic of cloud-edge cooperative computing vehicle-road-cloud communication

①远程驾驶端,利用 5G 网络高带宽、低时延的优势,视频可在 100 ms 以内呈现在远程驾驶员面前,驾驶员的控制指令在 20 ms 左右即可返回至远程驾驶车辆。

②网联自动驾驶车辆,这类车辆不仅具备自动驾驶功能,且可通过车对车(Vehicle to Vehicle, V2V)通信避免碰撞,通过车对基础设施(V2I)通信获取交通信号灯状态等信息,实现更高效、更安全的交通运行。其搭载的车载单元(On-board Unit, OBU)为智能网联车辆车载计算和通信单元,功能为进行车车通信以及与路侧单元进行车路通信。

③人工驾驶车辆(Human-driven Vehicle, HV),属于传统交通流中重要组成部分,具有一定

的通信能力,不具备自动驾驶能力,在任务卸载策略中不考虑该类型车辆。

④ 云端服务器功能为对计算密集型任务进行云端计算,即当计算任务数据量过大,边缘算力不足以支持边缘计算设备进行服务请求等。

⑤ 边缘端服务器主要部署于路侧设备(RoadSide Unit, RSU)中,其中RSU-A、RSU-B、RSU-a1、RSU-a2、RSU-a3、RSU-b1、RSU-b2、RSU-b3为边缘计算设备。其中,RSU-A、RSU-B为RSU中转基站,功能为对路侧单元、远程驾驶端以及云端信息进行通信转发,以及作为备用节点来接管计算任务;RSU-a1、RSU-a2、RSU-a3、RSU-b1、RSU-b2、RSU-b3为路侧单元,功能为进行边缘卸载计算。

⑥ 多源异构数据构成。智能网联车辆计算任务为在远程驾驶过程中的智能网联车端摄像头数据、方向盘转角、油门踏板开度、刹车信号、档位信号、娱乐设施等数据传输到远程驾驶端,以及进行数据备份等计算任务。

1.2 远程驾驶接管任务原理解析

本文将远程驾驶接管自动驾驶任务作为云边协同计算任务卸载策略中的主要研究对象,远程驾驶接管任务的核心流程依托云边协同计算环境,通过多阶段任务协同与实时决策,确保自动驾驶车辆在复杂场景下的安全接管。该流程涵盖数据采集、传输、决策与控制的全链路闭环,具体分为6个关键步骤,如图2所示。

步骤1:数据上传。由网联自动驾驶车辆车端搭载的多项传感器(摄像头、激光雷达和毫米波雷达等)获取多模态交通数据信息,开发驾驶舱端数据上传和接收程序,用以将驾驶舱端方向盘信号、油门踏板信号、制动踏板信号和转向灯信号实时上传到后台服务器(包含云端服务器、边缘端服务器和车端服务器),再转发到车端实时控制车辆。原则上需考虑

由车端数据处理产生的计算任务,但本文将研究重心设定至远程驾驶端的输出控制指令任务,故假设忽略此过程中车端产生的任务优先级和计算效率问题。

步骤2:数据传输。经由后台服务器将驾驶环境信息和车辆状态信息传输至通信服务器。

步骤3:车外视频数据推流。获取车辆多目摄像头数据,且包含音视频解码功能,将回传的音视频信号进行解码处理,并完成现场的画面及音频呈现,提供给远程驾驶人清晰的现场视角,从左至右分别输出左视、前视和右视驾驶视野画面。

步骤4:远程驾驶舱决策。由远程驾驶人通过传输回的画面做出是否需要接管决策,并相对应输出控制指令。

步骤5:驾驶数据信号转换。将驾驶舱端各控制信号协议和车端控制信号协议做匹配,用于模拟实际车辆控制的座舱,具备驱动、制动、转向执行机构,提供较为真实的驾驶环境和全面的车辆控制功能。在车载控制器中获取到驾驶信号后实现控车。

步骤6:最优决策输出。经由本文所设计的远程驾驶接管任务自适应卸载模型得到最优决策并输出控制,达成全局最低时延。

2 云边协同计算任务分类及卸载规则

本节首次提出面向远程驾驶接管场景的多维度任务分类与优先级动态绑定机制,突破了传统研究中任务类型与优先级分离的局限性。将任务划分为实时、计算密集和交互式3类,并结合紧急与一般两级优先级,形成动态绑定的任务管理体系。通过细粒度刻画任务的计算复杂度、资源占用和时延容忍度,通过预分类和精细分类提高任务分类准确度,为任务卸载提供了精准决策依据。提出了分层动态卸载规则,提出融合任务类型、优先级、网络带宽及边

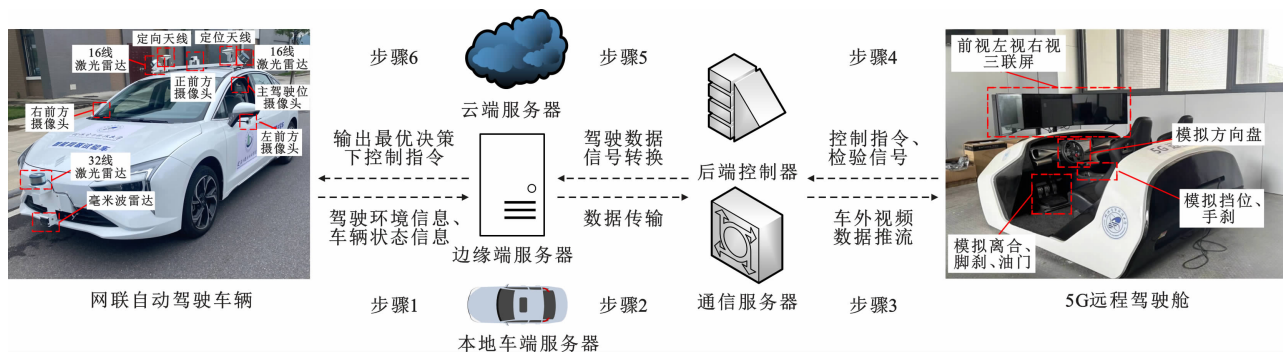


图2 远程驾驶接管任务原理

Fig.2 Remote driving takeover task principle

缘负载等多维变量的分层动态卸载规则。通过动态优先级调整、网络-算力联合优化(如带宽波动下的分段卸载)及备用节点断点续传机制,实现从“被动响应”到“主动适应”的范式转变。该框架突破了传统研究中边缘算力固定、通信场景单一的假设,支持复杂动态环境下的任务卸载优化,确保远程接管过程的低时延与高鲁棒性。最后,引入边缘负载动态感知与云端弹性资源协同机制,通过实时监控边缘负载波动,结合云端算力优势,优化任务分配策略。在边缘负载较高时,优先启用云端计算资源,解决现有研究默认边缘算力固定导致的资源利用率低问题,显著降低整体时延并提升卸载效率。

2.1 任务类型及优先级定义

在车联网环境的基础上结合云边协同计算特性,根据远程驾驶接管场景中任务的特性和计算需求,从计算复杂性、计算资源占用及时延容忍度等方面入手,定性分析各个指标特性,分为低、中、高3个等级。该分类不仅为任务卸载提供了精准的决策依据,更为后续构建动态规划模型奠定了结构化输入基础,使得模型能够针对任务特性与资源状态实现自适应匹配,从而显著提升系统可靠性。

将任务划分为实时任务、计算密集任务、交互式任务,并结合了紧急任务和一般任务2项优先级。所定义的任务分类如表1所示。

表1 任务分类

Table 1 Classification of tasks

任务类型	计算复杂度	计算资源占用	时延容忍度	任务优先处理端
实时任务	低	低	低	本地、边缘
计算密集任务	高	高	中	边缘、云
交互式任务	中	高	高	边缘、云

(1)实时任务:对时效性要求极高的任务,需在毫秒级内完成处理与响应。例如方向盘转角、油门踏板开度、刹车信号、档位状态等直接控制指令的传输与处理等。优先在本地或边缘节点处理,以最小化传输时延。

(2)计算密集任务:需要大量计算资源的任务,通常涉及复杂的数据处理与分析。例如多目摄像头实时画面传输、自动驾驶风险预判(如障碍物识别、轨迹规划)、高精度地图更新等。优先卸载至边缘节点或云端服务器,利用其高计算资源。

(3)交互式任务:涉及人机交互或非关键性后台服务的任务,对时延容忍度较高。例如人机对话、娱乐系统操作、非实时数据备份等。在边缘或云端资

源充足时卸载处理,避免占用本地计算资源。

经由任务分类处理后得到相对应的任务类型,依据任务紧急程度设置任务优先级为紧急任务和一般任务,当被判定为紧急任务则暂不考虑云端计算,优先本地或算力充足的边缘端运算。关于任务分类及优先级划分和任务端执行流程图如图3(以实时任务判别过程为例)所示。

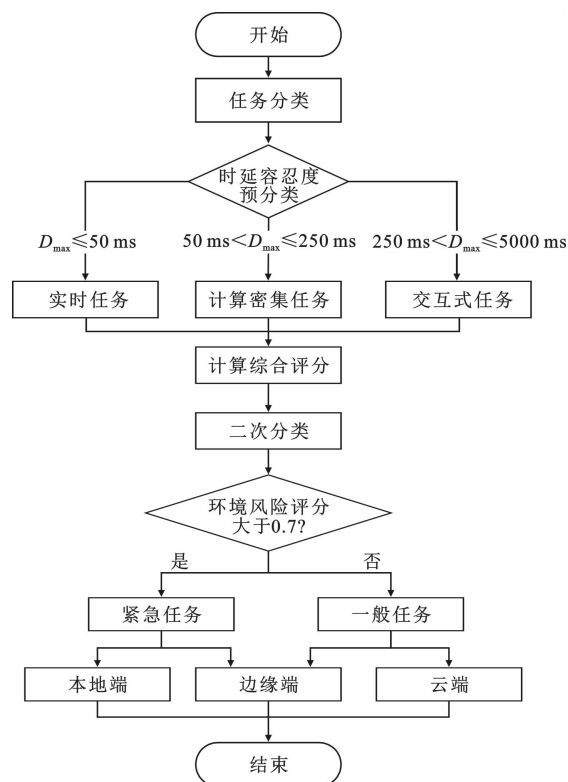


图3 任务分类及优先级判定过程

Fig. 3 Task classification and priority determination process

(1)紧急任务:系统中迫切需要立刻执行的任务,分配高优先级。例如接管请求任务、方向盘转角、油门踏板开度、刹车信号、档位信号、摄像头实时画面等数据传输任务。任务优先在本地和边缘端处理。

(2)一般任务:对计算资源需求较低且非紧急的任务,即使延后执行也可不受影响,例如人机对话、娱乐等任务,当遇突发任务例如远程驾驶任务接入,此类任务动态降低优先级并由车端卸载到边缘端和云端。

2.2 任务分类模型

以上任务类型分类定义设计了分类标准,即

$$T = f(Q, \rho, D) \quad (1)$$

式中: T 为任务类型; $f(Q, \rho, D)$ 为加权函数,其中 Q 为计算复杂度, ρ 为计算资源占用率, D 为时延容忍度。

为方便计算,将 Q 、 ρ 、 D 通过归一化处理,值越小表示需求越低。此外, $T = \{0, 1, 2\}$, 0 代表实时任务, 1 代表计算密集型任务, 2 代表交互式任务。

Q 主要由 CPU 频率、任务数据量、算法操作数等决定,第 i 个任务计算复杂度 Q_i 为

$$Q_i = \frac{\alpha_1 F_i}{F_{\max}} + \frac{\beta_1 M_i}{M_{\max}} + \frac{\gamma_1 S_i}{\tilde{S}} \quad (2)$$

$$\alpha_1 + \beta_1 + \gamma_1 = 1 \quad (3)$$

式中: F_i 为执行第 i 个任务时的 CPU 频率,即计算第 i 个任务所需的 CPU 周期数; $F_{\max}$ 为 CPU 最大频率; M_i 为计算第 i 个任务时的内存频率; $M_{\max}$ 为内存最大频率; S_i 为任务 i 数据量; $\tilde{S}$ 为任务大小基准值; α 为操作数权重; β 为数据量权重; γ 为硬件约束权重。

计算资源占用数据可由平台端直接获取,系统资源状态主要由边缘端服务器负载、网络带宽波动及本地算力限制决定,第 i 个任务计算资源占用 ρ_i 为

$$\rho_i = \frac{\alpha_2 F_i}{F_{\max}} + \beta_2 L(t) + \frac{\gamma_2 B(t)}{B_{\max}} \quad (4)$$

$$\alpha_2 + \beta_2 + \gamma_2 = 1 \quad (5)$$

式中: $B(t)$ 为 t 时刻的网络带宽波动情况; $B_{\max}$ 为带宽最大值,取 $500 \text{ Mb} \cdot \text{s}^{-1}$; $L(t)$ 为 t 时刻的边缘负载百分比。

任务时延容忍度由具体任务特性决定,例如无论是人工驾驶还是自动驾驶中,制动踏板和方向盘控制等底盘控制任务对时延容忍度极低时延容忍度 D 表示任务对处理延迟的敏感程度,可以通过任务的最大允许时延 $D_{\max}$ 与任务最小理论处理时延 $D_{\min}$ 来量化,即

$$D = \frac{D_{\max} - D_{\min}}{D_{\max}} \quad (6)$$

$D_{\min}$ 是指在理想条件下(网络波动稳定、算力充足),任务处理的理论下限时延。通常情况下其取值与本地总时延 D_1 一致,但其算力条件更优,为简化计算将二者取值保持一致,即

$$D_{\min} = D_1 = \frac{S_i}{C_1} \quad (7)$$

式中: C_1 为本地算力。

2.3 双阶段分类

本文所研究模型重点侧重于时延部分,因此将任务的时延容忍度要素放置首位,同时, $D_{\max}$ 为任务从触发到完成响应的最长时间阈值,超过此阈值可能导致系统失效或安全风险,因此在对任务分类前

可先进行预分类以达到更优分类准确性。

2.3.1 预分类

时延容忍度是远程驾驶接管任务中最关键的指标,需优先处理。根据历史数据及任务特性,预分类规则为

$$T' = \begin{cases} 0 & 0 < D_{\max} \leq 50 \text{ ms} \\ 1 & 50 \text{ ms} < D_{\max} \leq 250 \text{ ms} \\ 2 & 250 \text{ ms} < D_{\max} \leq 500 \text{ ms} \end{cases} \quad (8)$$

式中: T' 为任务预分类后的任务类型。

式(8)设定了 $D_{\max}$ 在 3 种区间下所属的 3 种任务分类,即在最大容忍时延分别在不同阈值下将任务分别分类为实时任务、计算密集型任务、交互式任务。其中,参考自标准《道路车辆功能安全》(ISO 26262)中所知,依据人类反应时间或控制闭环稳定性要求,通常为 $10 \sim 100 \text{ ms}$ (如自动驾驶紧急避障需不大于 50 ms),而人类可感知的延迟上限约为 400 ms ,由以上依据取其范围区间平均值为任务最大允许时延 $D_{\max}$ 的阈值。

2.3.2 二次分类

在预分类的基础上,通过加权综合评分进一步细化分类,从而可以得到任务分类综合评分 G 为

$$G = \omega_Q Q + \omega_\rho \rho + \omega_D D \quad (9)$$

$$\omega_Q + \omega_\rho + \omega_D = 1 \quad (10)$$

$$\omega_{\rho, \max} = 0.5 \quad (11)$$

$$\omega_{D, \max} = 0.8 \quad (12)$$

式中: ω_Q 、 ω_ρ 、 ω_D 分别为 Q 、 ρ 、 D 的权重因子; $\omega_{\rho, \max}$ 、 $\omega_{D, \max}$ 分别为 ω_ρ 、 ω_D 的最大值。

任务最终分类 T'' 为

$$T'' = \begin{cases} 0 & G \leq \theta_1 \cap T' = 0 \\ 1 & \theta_1 \leq G \leq \theta_2 \cap T' = 1, 0 < \theta_1 < \theta_2 \leq 1 \\ 2 & G \geq \theta_2 \cap T' = 2 \end{cases} \quad (13)$$

式中: θ_1 、 θ_2 均为参数,由历史数据训练调整,满足 $0 < \theta_1 < \theta_2 \leq 0$ 。

2.3.3 优先级绑定

采用二元优先级($P=0$ 表示紧急, $P=1$ 表示一般)与 3 类任务的强制映射关系为

$$P = \begin{cases} 0 & T'' = 0 \cup (T'' = 1 \cap E > \chi) \\ 1 & \text{其他} \end{cases} \quad (14)$$

式中: E 为环境风险评分(如碰撞概率、网络延迟); χ 为风险阈值,取值为 0.7 。

2.4 任务卸载规则

相较于传统基于固定阈值或单一指标的策略,任务卸载规则通过动态优先级调整、网络-算力联合优化及备用节点断点续传机制,提升了系统任务卸

载策略的主动性。这一创新不仅突破了现有研究对边缘算力的静态假设,更通过任务冲突消解与资源弹性分配,为复杂车联网环境下远程驾驶接管的低时延、高可靠卸载提供了理论闭环与实践路径。

2.4.1 卸载规则定义

针对现有研究中任务卸载策略静态化、边缘负载固化及通信场景单一的局限性,提出一种多维度动态协同卸载框架,将任务优先级、网络状态、边缘负载与云端弹性资源纳入统一决策体系,构建场景自适应的分层卸载规则如下。

(1)网络状况:网络带宽充足时,优先选择将任务卸载到边缘端和云端,网络状况较差时,任务优先在本地和边缘端处理。

(2)边缘端计算资源:边缘端计算资源充足时任务优先卸载到边缘端和云端,计算资源不足时,任务优先在本地和云端处理。

(3)备用节点:以 RSU-A 和 RSU-B 为备用节点,设置任务点回溯机制,可以在 RSU 路侧单元算力不足以及出现问题时,实现断点续传,以确保车辆系统的连续运行。

2.4.2 动态权重调整

动态权重调整机制旨在根据实时网络状态(如带宽波动)和边缘节点负载情况,自适应调整任务分类中 Q 、 ρ 和 D 的权重分配,以优化任务卸载决策。其核心逻辑为:高负载/低带宽时,优先保障时延敏感任务;低负载/高带宽时,兼顾计算效率与资源利用率。

(1)权重动态调整

初始权重为 ω_Q 、 ω_ρ 、 ω_D 且满足 $\omega_Q + \omega_\rho + \omega_D = 1$,根据实时状态动态更新权重 ω'_Q 、 ω'_ρ 、 ω'_D ,加入参数约束条件 ϕ 和 φ 。

(2)边界条件处理

针对网络带宽的波动情况,本文引入带宽敏感系数对时延权重进行实时修正,其更新规则为

$$\omega'_D = \omega'_D(t-1) + \phi \left(\frac{B_i}{B_{\max}} - 0.5 \right) \quad (15)$$

式中: ϕ 为带宽敏感系数, $\phi \in [0.2, 0.5]$ 。

式(15)意义在于:当带宽高于或低于均值时,相应的增加和减少,做到高任务卸载量和低时延的动态平衡。

考虑到边缘节点的实时负载压力,系统通过负载敏感系数动态调节任务分配权重,以规避过载风险,具体计算方式为

$$\omega'_\rho(t) = \omega'_\rho(t-1) + \varphi \left[\frac{L(t)}{L_{\max}} - 0.5 \right] \quad (16)$$

式中: φ 为负载敏感系数, $\varphi \in [0.3, 0.6]$; $L_{\max}$ 为最大负载,取 90%; $L(t)$ 为当前边缘节点负载率,满足 $0 \leq \frac{L(t)}{L_{\max}} \leq 1$ 。

式(16)意义在于:当边缘负载高于均值时,增加 ω_ρ 减少对过载节点的任务分配;当边缘负载低于均值时,减少 ω_ρ 鼓励利用空闲资源。

(3)权重归一化条件

使权重系数强制满足 $\omega'_Q + \omega'_\rho + \omega'_D = 1$,即

$$\omega'_Q = \max\{0, 1 - \omega'_D - \omega'_\rho\} \quad (17)$$

式(17)意义在于:确保 ω'_Q 非负,若剩余权重不足则置零。

(4)综合评分更新

任务最终分类评分 G' 为

$$G' = \omega'_Q Q + \omega'_\rho \rho + \omega'_D D \quad (18)$$

(5)动态卸载阈值设定

根据实时环境动态优化卸载阈值 δ 为

$$\delta = \hat{\delta} + \Delta \left[L(t) - \frac{B(t)}{B_{\max}} \right] \quad (19)$$

式中: $\hat{\delta}$ 为基础阈值,默认 $\hat{\delta} = 0.5$; Δ 为自适应调节权重,其物理意义在于量化状态反馈偏差对最终补偿量的贡献强度; $B(t)$ 为当前可用带宽,满足 $0 \leq \frac{B(t)}{B_{\max}} \leq 1$ 。

同时为阈值设定边界,避免所有任务卸载至云端及确保实时任务有最低阈值,令 $\delta_{\max}$ 、 $\delta_{\min}$ 分别为动态卸载阈值 δ 的最大和最小值。

(6)卸载策略生成

卸载动作函数 $\pi = \{0, 1, 2\}$,分别代表本地处理、边缘卸载、云端卸载,根据 G' 和任务优先级 P 生成

$$\pi = \begin{cases} 0 & P=0 \\ 1 & P=1 \cap G' \leq \delta \\ 2 & P=1 \cap G' > \delta \end{cases} \quad (20)$$

3 基于动态规划的任务自适应卸载模型

基于第 2 节构建的云边协同计算环境以及任务卸载规则,进行任务自适应模型的构建。动态规划算法可以有效地解决具有最优子结构性质和重叠子结构性质的问题,即是一种可以高效的解决多阶段决策过程优化问题的数学方法,可以通过避免重复计算、确保全局最优解、减少算法的时间复杂度和空间复杂度。

本节研究的云边协同自适应卸载模型可以表示为多阶段决策问题,来优化计算任务卸载时延,即寻找任务卸载最短时延,因此使用动态规划算法来构建任务自适应卸载模型。

3.1 确定模型空间

任务优先级分为2种,紧急任务和一般任务,即任务优先级可以表示为 $P=\{0,1\}$,任务可以在本地或者卸载到边缘和云端进行计算,即卸载动作可以定义为 $\pi=\{0,1,2\}$,车端、边缘和云端算力可以表示为 C_1, C_2, C_3 。

3.2 确定子问题状态

将问题定义为子问题的集合,每个子问题可以表示为 $V(i, j)$,其中 i 为任务索引, i 的取值范围为 $[1, 100]$,任务索引 i 与时间步长 t 严格同步对齐,每个时间步长 t 对应处理第 i 个任务; j 为可卸载的服务器单元, j 的取值范围是 $[1, 3]$ 。

3.3 确定状态转移方程

系统内任务的状态转移方程为

$$V(i, j) = \max\{V(i, j), V(i-1, j) + r(i, j)\} \quad (21)$$

式中: $V(i, j)$ 为前 i 个任务分配到 j 个服务器进行处理所获得的最大奖励; $r(i, j)$ 为任务 i 分配到第 j 个服务器上所获得的卸载时延奖励。

由卸载规则可知,当任务为紧急任务,优先在本地和边缘进行处理,当任务为一般任务,优先在边缘端和云端进行处理。

由于本地计算通信有线连接,默认忽略传输时延,因此只考虑计算时延,即

$$D_1 = \frac{S_i}{C_1} \quad (22)$$

边缘计算端既需要考虑计算时延,还需考虑动态的边缘负载,也需要考虑传输时延,即

$$D_2 = \frac{S_i}{L(t)} \left(\frac{1}{C_2} + \frac{1}{B} \right) \quad (23)$$

式中: D_2 为边缘计算总时延。

而云端算力资源充足但距离较远,因此考虑云端时延时忽略计算处理时延,只考虑传输时延,即

$$D_3 = \frac{S_i}{C_3} + \tilde{D}_3 \quad (24)$$

式中: D_3 为云端总时延; $\tilde{D}_3$ 为云端传输时延,即从云端传输信息至边缘端或车端所需的时间,由云端距离及网络带宽决定。

任务的总时延 D_T 为

$$D_T = \sum_{i=1}^n D_i \quad (25)$$

由于本地、边缘和云端计算的时延数值可能差别较大,因此,为了更好地对不同卸载方式的时延进行衡量及表征,设计奖励函数 R ,对总时延加1(防止分母为0)取倒数,即

$$R = \frac{1}{D_T + 1} \quad (26)$$

由于总时延在分母位置,且该奖励函数仅考虑任务时延单一变量,函数值越大,即总时延越小,表示当前任务卸载方案越好。

3.4 初始化边界条件

$V(0, j) = 0$ 表示没有任务时的最优解为0; $V(i, 0) = 0$ 表示没有可用边缘端服务器时的最优解为0。

3.5 计算 $V(i, j)$ 的值

根据状态转移方程和边界条件计算出每一个状态的值。在本模型中,每一个状态表示当前剩余任务和可用边缘端服务器的数量,以及已经分配的任务和边缘端服务器之间的匹配关系。使用状态转移方程来计算每一个状态的最大奖励值。

3.6 回溯寻优

回溯是指从最优状态开始,逆向推导出最优的任务分配方案,具体步骤如下。

步骤1:确定最优值位置,通过比较不同状态的值来找到最优解。找到每一个任务在不同边缘端服务器上所获得的最大奖励值,以确定最优的任务分配方案,即任务 i 在不同服务器上所获得最大奖励值。

步骤2:初始化最优路径,记录每一个状态对应的最优选择。记录每一个任务分配状态所对应的最优边缘端服务器选择,以便回溯时使用。

步骤3:回溯,从最优状态开始,根据状态转移方程和最优路径,依次向前推导,直到初始状态为止,即可得到最优的任务分配方案。根据最优路径,逐步恢复出每一个任务分配状态下的最优边缘端服务器选择,从而实现任务自适应卸载,即从 $V(i, j)$ 开始,根据 $V(i, j)$ 的值和状态转移方程,依次向前推导,直到 $V(1, 1)$ 为止,即可得到最优的任务分配方案,从而实现任务自适应卸载。

4 复杂度分析与仿真实现

本节首先对任务自适应卸载算法进行复杂度分析;其次,基于实际场景需求构建包含任务特性、算力资源及网络状态的多维度数据集,确保数据的多

样性与覆盖性;随后,设计对比试验方案,涵盖本地、边缘与云端卸载的组合策略,并引入边缘负载变量以模拟算力波动;最后,通过奖励函数量化时延优化

效果,结合不同算力条件下的试验结果,验证模型在动态环境中的鲁棒性与适应性。整体试验流程图及仿真所用的硬件、软件配置如图 4 所示。

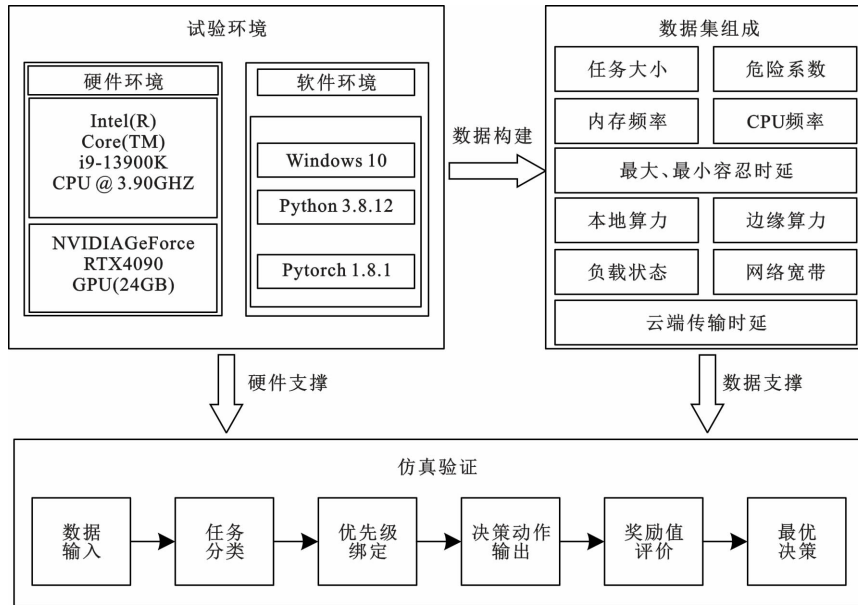


图 4 试验流程

Fig. 4 Experimental flow

4.1 算法复杂度分析

4.1.1 时间复杂度分析

动态规划算法的时间复杂度主要取决于状态转移方程中的空间规模与状态转移次数。定义模型参数有任务数量 N , 可卸载的计算节点数量为固定节点数 3 (本地、边缘、云端), 状态定义 $V(i, j)$ 表示前 i 个任务分配到 j 个计算节点的最大累计奖励值。状态转移过程中, 每个任务需遍历所有可能的计算节点以确定最优卸载策略, 即求最大值。第 i 层状态严格依赖第 $i-1$ 层状态, 形成链式递推结构, 对于第 i 个任务, 需计算 3 次转移操作。因此总状态数为 $N \times 3$, 每个状态需执行 k 次比较和更新操作。时间复杂度可表示为 $O(N \times 3^2)$, 可简化为 $O(N)$ 。此外, 奖励函数调用次数为 $N \times 3 \times 3 = 9N$, 比较操作次数为 $N \times 3 \times 3 = 9N$, 加法操作次数为 $N \times 3 \times 3 = 9N$, 所以操作总量为 $27N$ 。以上分析表明算法随任务数量线性增长, 具备较高的计算效率, 适用于实时性要求较高的远程驾驶接管场景。

4.1.2 空间复杂度分析

算法的空间复杂度由状态存储需求决定, 其核心在于管理任务分配过程中的中间状态。原始设计需维护一个二维数组 $V(N, 3)$ 储存各状态的最优奖励值, 其空间消耗为 $O(N \times 3)$, 同理, 空间复杂度可简化为 $O(N)$ 。该矩阵维度为 $N \times 3$, 每个元素存储

浮点型状态值, 原始空间需求为 $12N$ 字节。同时需存储决策路径矩阵 (记录最优节点选择), 维度为 $N \times 3$ 的整型数据 (占用 $3N$ 字节), 总空间消耗达 $15N$ 字节, 对应 $O(N)$ 空间复杂度。表明算法在内存占用方面具有可扩展性, 能够适应大规模任务卸载需求。

4.2 多维数据集构建

参考现有研究中涉及的对自动驾驶任务的计算需求分类方法^[17], 5G 环境中采用的经典 C-V2X 通信带宽范围^[16]、自适应卸载算法任务二元分类^[21]、实际场景边缘算力动态波动特性^[24]、多资源卸载能耗模型设定边缘节点负载阈值范围^[20]等研究, 依据任务大小、CPU 频率、内存频率、边缘负载、本地算力、边缘算力、信道带宽、云端传输时延、任务最大和最小容忍时延、风险系数共 11 个数据类型构建云边计算卸载数据集, 见表 2。

任务大小设定范围为 $[20, 100]$ Mb, 对不同任务大小输入通过任务分类后绑定不同优先级, 0 代表紧急任务, 1 代表一般任务, 带宽范围为 $[300, 500]$ $\text{Mb} \cdot \text{s}^{-1}$, 边缘算力区间为 $[50, 100]$ TOPS, 云端传输时延由本地算力和比边缘算力确定。

4.3 试验与结果分析

为了便于比较, 本文考虑以下 2 种对比方案。

方案 1: 不允许云端执行任务卸载。仅允许在

表 2 数据集参数说明
Table 2 Dataset parameter description

参数类别	符号	取值范围/类型
任务特性	S_i/Mb	$[20,100]$
	F_i/MHz	$[30,100]$
	M_i/MHz	$[30,100]$
网络条件	$B/\text{Mb} \cdot \text{s}^{-1}$	$[300,500]$
资源 状态	C_1/TOPS	$[10,50]$
	C_2/TOPS	$[50,100]$
	C_3/TOPS	$[0,+\infty]$
	$L/\%$	$[20,80]$
	$\tilde{D}_3/\text{ms}$	由 C_1, C_2, B 和传输距离计算得到, 传输距离设定为 $[1,2\ 000]\text{ m}$

车端和边缘端执行任务卸载策略,并按是否考虑边缘端服务器动态负载状态对比试验,此组试验目的是考虑真实的车联网环境下边缘端服务器负载动态变化对本文任务卸载模型整体优化的检验。

方案 2:允许云端执行任务卸载。在方案 1 的基础上,允许云端执行任务卸载策略,此组试验既可以检验云端是否可以在边缘端负载过高时提供一种可靠的备用方案,这可以为远程驾驶接管提供良好的基础,同时又可以检验在非紧急状态下对全局任务卸载的优化作用。

4.3.1 仅允许边缘端执行任务卸载

本节试验仅允许任务在本地端和边缘端执行,暂不加入云端,同时分为动态边缘负载和固定边缘负载值两组对比时延。图 5 为仅允许本地端和边缘端执行任务时,考虑动态负载和固定负载的任务卸载奖励曲线。考虑到样本数量,为更直观对比结果,将样本数量平均分为 4 份,即每个分图代表其中 25 个任务卸载奖励曲线,累加二者的面积差可得两曲线最终输出的总奖励值大小。经数据验证可得,全局奖励值提升约 13.2%,局部奖励值提升最高可达 58.2%。

值得注意的是,需要对出现黄色负增益的部分加以分析。通过时间序列对齐,可发现黄色部分对应的 $L(t)$ 值偏大,即边缘端服务器处于高负载状态,而此方案下不允许云端的介入,任务无法卸载至高负载下的边缘端,只能强行在本地执行,因此产生的延迟相对会大;反观在不考虑负载动态变化的条件下,固定阈值会营造一种边缘端服务器始终保持良好的负载状态的假象,因此允许更多任务进入边缘端从而获得更低延迟,这与真实车联网环境相悖,过量任务涌入易造成服务器宕机破坏全局利益。因此本文模型在考虑并模拟驾驶环境中的边缘端服务

器负载情况下,避免了陷入局部最优情况,提升了全局任务卸载和资源调度优化能力。

4.3.2 允许云端执行任务卸载

本节试验允许云端加入任务卸载行列,检验其对边缘端计算资源补充作用,以及对全局任务卸载总时延的优化作用。

图 6 在加入了云端的海量计算资源后,系统内整体算力有了极大的提升,由于 2 组试验共用一种数据集,因此在奖励值曲线整体上有相似的走向和趋势,尤其是在模型赋予了任务相同的优先级和执行动作的情况下。此外,在允许云端进入的情况下试验增加了网络系数用以区分当前网络状态是否满足云端数据高速的互传。

由于与方案 1 得到的奖励值曲线存在相似的趋势,某些特征在云端的作用下被放大,例如在同一时间步长端内的正增益扩大,即绿色区域的面积有明显增加,这是由于云端的算力带来的直接影响。数据验证可知,在加入云端后,方案 2 整体奖励值较方案 1 有较大提升,其中在固定负载阈值方案中奖励值提升约 21.7%,在动态负载方案中奖励值提升约 23.6%;此外方案 2 中考虑动态负载方案较固定负载方案奖励值提升约 14.9%。

加入云端后随之带来的变化还有黄色负增益的增加,经数据分析后得出造成原因与方案 1 相似,但多了一种成因。在边缘高负载下任务无法卸载至边缘端,根据卸载规则任务会根据其绑定的优先级选择本地或云端执行,如选择在本地执行则与上述一致,在方案 2 中多了卸载到云端的选择,如在卸载至云端时遇网络波动也会造成较大延迟,从而影响任务卸载奖励值获取。从数据样本数来看后者属于小概率事件,但在真实驾驶环境下网络环境复杂,波动幅度和频率较大,此类情况不可完全避免。

由于方案整体奖励值有所提升,且遇突发状态(如网络延迟波动,边缘端服务器过载)下依然可将任务分配至合适平台并减少全局时延,证明了该模型具备一定适应性和鲁棒性。

4.3.3 试验结果分析

基于第 4.3.1 节和第 4.3.2 节试验结果,从卸载时延奖励的角度,通过试验进行远程驾驶接管过程中,基于云边协同计算的系统时延优化技术研究的验证与分析,得到以下试验结果。

(1)自适应卸载策略显著降低时延:在仅允许在考虑边缘端服务器负载动态变化时,本研究的自适应卸载策略得到的全局奖励值提升约 13.2%;在允

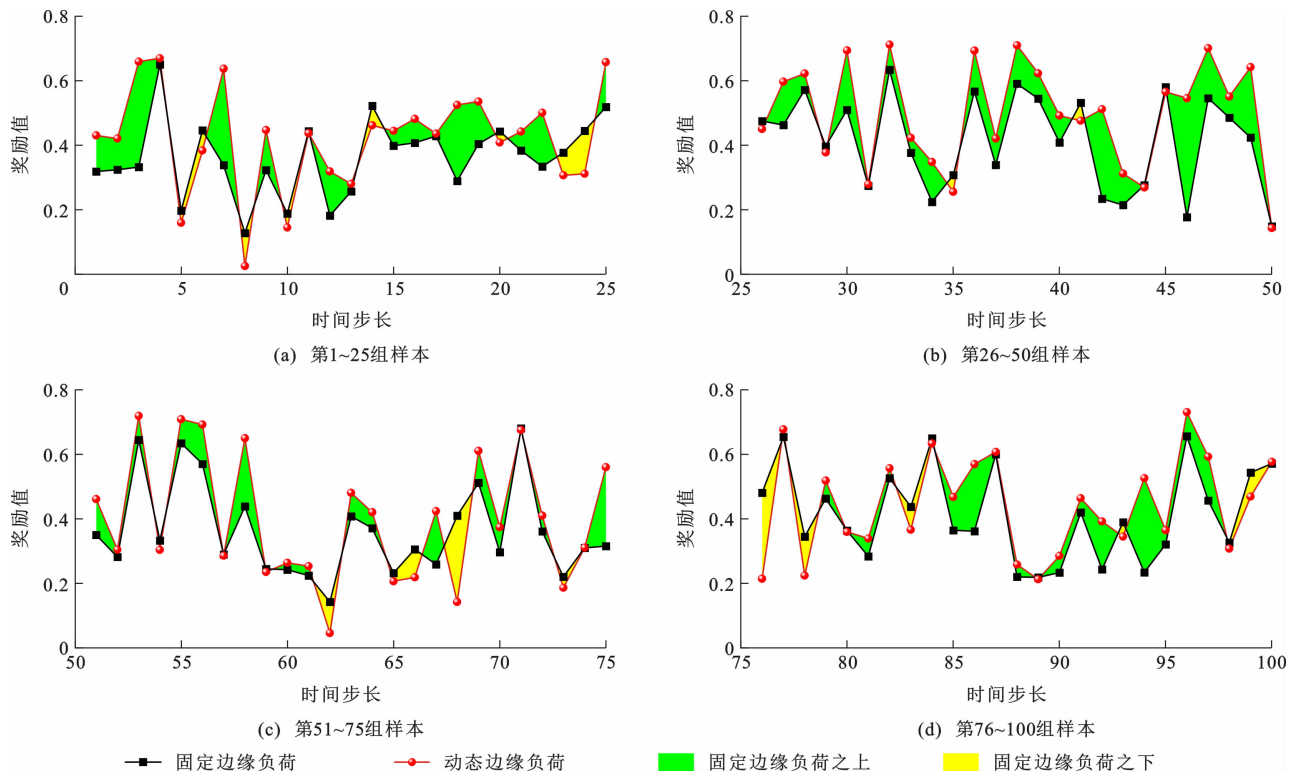


图 5 仅允许边缘端执行任务卸载

Fig. 5 Allow only the edge to perform task offloading

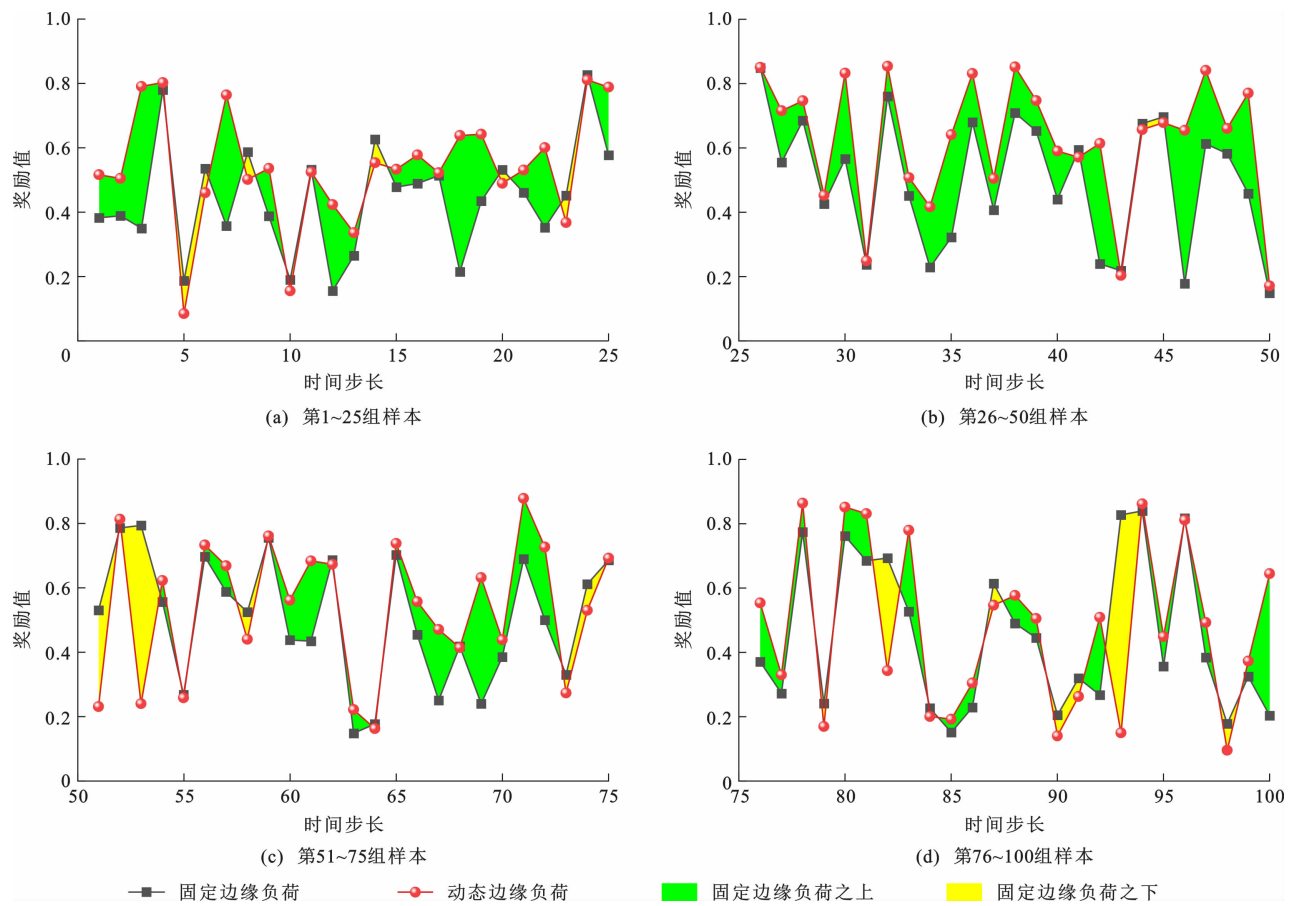


图 6 允许云端执行任务卸载

Fig. 6 Allow cloud-based task offloading

许云端接入进行任务卸载后,全局奖励值提升约18.1%。这表明模型能根据边缘负载状态差异动态优化任务分配,确保全局时延最小化。

(2)突出云端协同计算的显著优势:引入云端资源后,在动态边缘负载场景下,云边协同的奖励值较单一边缘计算提升约23.6%,充分体现了云端弹性资源对时延敏感任务的补充作用。尤其在本地算力不足时,云端计算有效缓解了边缘负载压力,避免了时延激增。

(3)边缘负载的动态适应性提升:模型通过实时感知边缘负载波动,优先启用云端资源,解决了传统研究中默认边缘算力固定导致的资源利用率低问题。试验表明,在边缘负载较高时,自适应策略的时延稳定性显著优于规则卸载。

(4)复杂场景下的鲁棒性提升:在动态网络带宽 $[300, 500] \text{ Mb} \cdot \text{s}^{-1}$ 及任务优先级冲突场景下,模型通过分层动态卸载规则(备用节点断点续传、网络-算力联合优化)实现了从“被动响应”到“主动适应”的范式转变,验证了其在复杂车联网环境中的可靠性。

综上,试验结果从时延优化、负载适应性和云端协同三方面证明了所提模型的有效性,为远程驾驶接管的低时延、高可靠性需求提供了理论与技术支撑。

5 结 语

(1)本文针对智能网联车辆远程驾驶接管过程中存在的任务卸载问题,提出了一种基于云边协同的自适应卸载模型。该模型通过构建云边协同计算环境,综合考虑任务类型、任务优先级、网络状况和边缘端计算资源等因素,设计了灵活的任务卸载规则。能够根据不同的任务需求和计算资源状况,自适应地进行任务卸载,有效降低了系统整体时延,提升了远程驾驶接管的可靠性。为智能网联车辆在复杂交通环境下的任务卸载提供了一种新的解决方案,保障了远程驾驶接管过程的稳定性和安全性。

(2)通过动态规划算法,构建了任务自适应卸载模型,并利用自建数据集进行了试验验证。结果表明,在不同算力条件和边缘负载状态下,该模型均能有效降低任务卸载时延,相较于传统的规则卸载方法,具有更好的性能和更高的鲁棒性。特别是在引入云端计算资源后,进一步优化了任务卸载策略,证明了云边协同计算在提升远程驾驶接管可靠性方面的显著优势。试验仿真结果验证了模型的有效性,

为实际应用提供了有力支持。

(3)本研究为任务卸载过程中的安全性问题(如数据隐私保护与网络安全防护)提供了初步框架,其具体机制与防护策略值得在未来工作中深入探索,后续可进一步融合动态交通场景的复杂性,并探索模型在5G/6G混合组网等异构网络环境下的适应性。将模型部署至真实智能网联车辆系统中进行验证与优化,也是一个具有重要应用价值的方向。此外,结合网络安全领域的最新进展,构建跨领域协同的安全增强型任务卸载机制,将有助于实现智能网联车辆远程驾驶接管在效率与可靠性方面的进一步提升。

参考文献:

References:

- [1] 秦严严,罗钦中,贺正冰. 网联自动驾驶车辆混合交通流专用道管控方法[J]. 交通运输工程学报, 2023, 23(3): 221-231.
QIN Yan-yan, LUO Qin-zhong, HE Zheng-bing. Management and control method of dedicated lanes for mixed traffic flows with connected and automated vehicles[J]. Journal of Traffic and Transportation Engineering, 2023, 23(3): 221-231.
- [2] 上官伟,李鑫,柴琳果,等. 车路协同环境下混合交通群体智能仿真与测试研究综述[J]. 交通运输工程学报, 2022, 22(3): 19-40.
SHANGGUAN Wei, LI Xin, CHAI Lin-guo, et al. Research review on simulation and test of mixed traffic swarm in vehicle-infrastructure cooperative environment[J]. Journal of Traffic and Transportation Engineering, 2022, 22(3): 19-40.
- [3] 张毅,裴华鑫,姚丹亚. 车路协同环境下车辆群体协同决策研究综述[J]. 交通运输工程学报, 2022, 22(3): 1-18.
ZHANG Yi, PEI Hua-xin, YAO Dan-ya. Research review on cooperative decision-making for vehicle swarms in vehicle-infrastructure cooperative environment[J]. Journal of Traffic and Transportation Engineering, 2022, 22(3): 1-18.
- [4] ZHAO H Z, WU H, LU N N, et al. Lane changing in a vehicle-to-everything environment: Research on a vehicle lane-changing model in the tunnel area by considering the influence of brightness and noise under a vehicle-to-everything environment[J]. IEEE Intelligent Transportation Systems Magazine, 2023, 15(2): 225-237.
- [5] 赵红专,张继康,潘佳雯,等. 基于改进视觉算法的自动驾驶风险预判模型[J]. 交通运输系统工程与信息, 2024, 24(5): 79-90, 139.
ZHAO Hong-zhuan, ZHANG Ji-kang, PAN Jia-wen, et al. Automatic driving risk prediction model based on improved vision algorithm[J]. Journal of Transportation Systems Engineering and Information Technology, 2024, 24(5): 79-90, 139.

- [6] HOU C, ZHAO Q C. Optimal task-offloading control for edge computing system with tasks offloaded and computed in sequence[J]. *IEEE Transactions on Automation Science and Engineering*, 2023, 20(2): 1378-1392.
- [7] 王新胜, 卞震. 基于贝叶斯模型的驾驶行为识别与预测[J]. *通信学报*, 2018, 39(3): 108-117.
WANG Xin-sheng, BIAN Zhen. Driving behavior recognition and prediction based on Bayesian model[J]. *Journal on Communications*, 2018, 39(3): 108-117.
- [8] SATRIA D, PARK D, JO M. Recovery for overloaded mobile edge computing[J]. *Future Generation Computer Systems*, 2017, 70: 138-147.
- [9] DUAN J R, REN K, ZHOU W, et al. A service migration method for resource competition in mobile edge computing[C]//*IEEE. 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. New York: IEEE, 2021: 1-8.
- [10] JITANI A, MAHAJAN A, ZHU Z W, et al. Structure-aware reinforcement learning for node-overload protection in mobile edge computing[J]. *IEEE Transactions on Cognitive Communications and Networking*, 2022, 8(4): 1881-1897.
- [11] 李晨, 严新平, 刘佳仑, 等. 船舶远程驾驶控制系统设计与应用[J]. *交通运输工程学报*, 2024, 24(5): 333-347.
LI Chen, YAN Xin-ping, LIU Jia-lun, et al. Design and application of ship remote-driving control system[J]. *Journal of Traffic and Transportation Engineering*, 2024, 24(5): 333-347.
- [12] 张继康. 基于自动驾驶风险预判的远程驾驶接管和云边协同可靠性提升方法研究[D]. 桂林: 桂林电子科技大学, 2024.
ZHANG Ji-kang. Remote driving takeover and cloud-edge cooperative reliability improvement method based on autonomous driving risk prediction[D]. Guilin: Guilin University of Electronic Technology, 2024.
- [13] 郭延永, 刘佩, 袁泉, 等. 网联自动驾驶车辆道路交通安全研究综述[J]. *交通运输工程学报*, 2023, 23(5): 19-38.
GUO Yan-yong, LIU Pei, YUAN Quan, et al. Review on research of road traffic safety of connected and automated vehicles[J]. *Journal of Traffic and Transportation Engineering*, 2023, 23(5): 19-38.
- [14] MACH P, BECVAR Z. Mobile edge computing: A survey on architecture and computation offloading[J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(3): 1628-1656.
- [15] 陈玉平, 刘波, 林伟伟, 等. 云边协同综述[J]. *计算机科学*, 2021, 48(3): 259-268.
CHEN Yu-ping, LIU Bo, LIN Wei-wei, et al. Survey of cloud-edge collaboration[J]. *Computer Science*, 2021, 48(3): 259-268.
- [16] XU X L, ZHANG X, LIU X H, et al. Adaptive computation offloading with edge for 5G-envisioned Internet of connected vehicles[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 22(8): 5213-5222.
- [17] DAI H J, ZENG X Y, YU Z L, et al. A scheduling algorithm for autonomous driving tasks on mobile edge computing servers[J]. *Journal of Systems Architecture*, 2019, 94: 14-23.
- [18] KUANG Z F, MA Z H, LI Z, et al. Cooperative computation offloading and resource allocation for delay minimization in mobile edge computing[J]. *Journal of Systems Architecture*, 2021, 118: 102167.
- [19] CHEN X H, CAI Y L, LI L Y, et al. Energy-efficient resource allocation for latency-sensitive mobile edge computing[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(2): 2246-2262.
- [20] WEI Z, YU X B, ZOU L. Multi-resource computing offload strategy for energy consumption optimization in mobile edge computing[J]. *Processes*, 2022, 10(9): 1762.
- [21] HUANG P, DENG M J, KANG Z L, et al. Self-adaptive learning of task offloading in mobile edge computing systems[J]. *Entropy*, 2021, 23(9): 1146.
- [22] XU F, XIE Y, SUN Y Y, et al. Two-stage computing offloading algorithm in cloud-edge collaborative scenarios based on game theory[J]. *Computers and Electrical Engineering*, 2022, 97: 107624.
- [23] 唐宁昆, 王高才. 云边协同环境下一种面向任务的多智能体强化学习任务卸载方案[J]. *小型微型计算机系统*, 2025, 46(4): 863-875.
TANG Ning-kun, WANG Gao-cai. Task-oriented multi-agent reinforcement learning task offloading scheme for cloud-edge computing[J]. *Journal of Chinese Computer Systems*, 2025, 46(4): 863-875.
- [24] ZHAO J H, LI Q P, GONG Y, et al. Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(8): 7944-7956.
- [25] LAI S W, ZHAO R, TANG S P, et al. Intelligent secure mobile edge computing for beyond 5G wireless networks[J]. *Physical Communication*, 2021, 45: 101283.
- [26] BUTT U A, AMIN R, MEHMOOD M, et al. Cloud security threats and solutions: A survey[J]. *Wireless Personal Communications*, 2023, 128(1): 387-413.