

星地协同中基于多智能体的时敏任务调度优化策略

陈娟^{1,2}, 钟杰¹, 吴宗玲³, 田谛¹, 陈玉杰¹

(1. 西华大学计算机与软件工程学院, 四川 成都 610039; 2. 云南财经大学云南省服务计算重点实验室, 云南 昆明 650221;
3. 西南交通大学信息科学与技术学院, 四川 成都 611756)

摘要: 随着智能物联技术与5G/6G通信技术的深度融合, 卫星边缘计算(SatEC, satellite edge computing)凭借空天协同计算网络, 为地面网络覆盖薄弱区域提供了新型算力服务。然而, SatEC系统面临星地动态资源分配失衡与多维时空约束下任务优先级控制不足的双重挑战。现有方法在分层决策、时空特征提取及任务紧急度量映射方面存在缺陷, 导致时敏任务处理效率受限。为此, 提出了一种基于自注意力时间卷积网络的多智能体深度强化学习算法。该算法通过构建多智能体架构实现任务优先级排序与资源分配的联合优化, 采用融合时空特征的混合神经网络精准提取星地协同场景的动态关联特性, 并建立基于概率模型的动态调度机制, 协同优化时延约束与任务完成率。仿真结果表明, 相较于基准算法, 该算法在任务完成率与时延控制方面均实现了显著的提升, 验证了其在复杂卫星边缘计算场景中的有效性与优越性。

关键词: 卫星边缘计算; 资源分配; 任务优先级; 自注意力时间卷积网络; 多智能体深度强化学习

中图分类号: TP393.08

文献标志码: A

doi: 10.11959/j.issn.2096-3750.2026.00507

Multi-agent-based time-sensitive task scheduling optimization strategy in satellite-terrestrial collaboration

Chen Juan^{1,2}, Zhong Jie¹, Wu Zongling³, Tian Di¹, Chen Yujie¹

1. School of Computer and Software Engineering, Xihua University, Chengdu 610039, China

2. Yunnan Key Laboratory of Service Computing, Yunnan University of Finance and Economics, Kunming 650221, China

3. School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China

Abstract: With the deep integration of intelligent Internet of things technology and 5G/6G communication technology, satellite edge computing (SatEC) offers new computational services to areas with weak terrestrial network coverage through its aerospace collaborative computing network. However, the SatEC system faces dual challenges of unbalanced dynamic resource allocation between satellite and ground and insufficient task priority control under multi-dimensional spatiotemporal constraints. Existing methods have defects in hierarchical decision-making, spatiotemporal feature extraction, and task urgency quantification mapping, which limit the efficiency of time-sensitive task processing. To address this problem, a multi-agent deep reinforcement learning algorithm based on self-attention temporal convolutional networks was proposed in this paper. The algorithm achieved joint optimization of task prioritization and resource allocation by constructing a multi-agent architecture, employed a hybrid neural network integrating spatiotemporal features to accurately extract dynamic correlation characteristics of satellite-ground collaboration scenarios, and established a dynamic schedul-

收稿日期: 2025-04-16; 修回日期: 2025-06-22

通信作者: 吴宗玲, zl_wu@swjtu.edu.cn

基金项目: 四川省网络文化研究中心课题 (No. WLWHZX-09); 四川省重点实验室服务科学与创新开放项目 (No. KL2411); 云南省服务计算重点实验室开放课题 (No. YNSC24118)

Foundation Items: The Project of Sichuan Network Culture Research Center (No. WLWHZX-09), the Open Project of Key Laboratory Service Science and Innovation of Sichuan Province (No. KL2411), the Foundation of Yunnan Key Laboratory of Service Computing (No. YNSC24118)

ing mechanism based on a probabilistic model to synergistically optimize latency constraints and task completion rates. Simulation results show that, compared with the baseline algorithm, the proposed algorithm achieves significant improvements in both task completion rate and delay control, demonstrating its effectiveness and superiority in complex satellite edge computing scenarios.

Key words: SatEC, resource allocation, task priority, self-attention temporal convolutional network, multi-agent deep reinforcement learning

0 引言

随着智能物联技术与5G/6G通信技术的融合演进,全球数据总量呈现爆发式增长,传统云计算架构在应对实时数据处理需求时逐渐显现出核心性能瓶颈^[1]。作为新型分布式计算范式,卫星边缘计算(SatEC, satellite edge computing)通过近地轨道(LEO, low earth orbit)卫星构建空天协同计算网络,为地面网络覆盖薄弱区域(如远洋、极地、荒漠)提供新型算力服务模式^[2]。相较于地面边缘计算节点, SatEC系统凭借星间链路的广域覆盖特性,可突破地理环境对算力部署的限制,在应急救援、环境监测、智慧交通等领域展现出独特的应用价值^[3]。

然而,当前 SatEC系统面临着双重技术挑战:第一,星地动态协同计算须解决资源分配的最优决策问题,本地终端计算与星载算力卸载的失衡易导致任务堆积或资源争用^[4];第二,多维时空约束下的任务优先级控制机制亟待优化,尤其是在自动驾驶、灾害预警等时敏场景中,任务超时处理可能引发系统性风险^[5]。

为了解决上述问题,研究者在该领域已经做出了许多贡献。例如,文献[6-8]提出了基于博弈论的卫星任务卸载算法,有效地缓解了多用户设备资源竞争问题;文献[9-11]通过凸优化理论构建联合迭代算法,显著地降低了多卫星系统的能量及时延开销。然而,这些方法依赖精确的数学模型和完整的环境信息,在卫星网络动态拓扑变化中易出现适应性不足的问题^[12]。尽管目前的许多启发式算法,例如,遗传算法^[13]、粒子群优化^[14]、模拟退火^[15]等优化方法已经克服了传统方法对精确模型和全局信息的依赖,但在面对多任务并发冲突、资源状态异步更新等复杂场景时,仍存在决策滞后性问题,难以满足时空敏感型任务对实时决策与弹性调度的双重需求。

近年来,深度强化学习(DRL, deep reinforce-

ment learning)因其优秀的自适应能力成为卫星边缘计算领域的研究热点。例如,文献[16]使用Q-learning算法优化了空地一体化网络中的数据收集时间;文献[17]采用双深度Q学习(DDQL, double deep Q-learning)算法,有效地减少了移动边缘计算辅助空地车辆网络中的计算和通信成本;文献[18]提出了一种基于DRL的任务卸载(DRTO, DRL-based task offloading)框架,在降低整体系统时延和增强用户体验方面显示出了巨大的潜力;文献[19]提出了一种多分支深度Q网络(BDQN, multi-branch deep Q network)算法,用于时变边缘计算环境,显著地降低了任务时延和失败率。然而,现有的DRL方法^[16-22]存在3个根本性缺陷:首先,多数研究采用单智能体架构,难以刻画任务优先级和资源分配的分层式决策特性;其次,传统状态表征方法(如全连接神经网络)对星地协同场景的时空关联特征提取能力不足,导致策略网络性能低下;最后,奖励函数设计未能建立任务紧急度与时空资源的量化映射关系,这种缺陷会使时敏任务的超时率上升。

针对上述技术瓶颈,本文提出了一种基于自注意力时间卷积网络的多智能体深度强化学习(SATCN-MADRL, multi-agent deep reinforcement learning algorithm based on self-attention temporal convolutional network)算法,旨在有效提升 SatEC系统的调度性能。本文的主要贡献包含以下3方面。

1) 构建了适用于 SatEC系统的多智能体深度强化学习架构,实现了任务优先级排序与资源分配的联合优化,有效地解决了星地动态协同计算中的资源分配最优决策问题。

2) 设计了独特的融合时空特征的混合神经网络模型,该模型结合自注意力机制与时间卷积网络,能够精准地提取星地协同场景中的时空关联特征,为任务优先级控制提供了强大的数据支撑。

3) 建立了基于概率模型的动态调度机制,并构建了时延约束与完成率协同优化的目标函数,显著

地提升了 SatEC 系统在高维时空约束下的任务优先级控制能力。仿真结果表明，相较于基准算法，本文所提方案在任务完成率、时延控制等关键指标上有显著提升。

1 系统模型及问题描述

SatEC 星地协同架构如图 1 所示，SatEC 框架采用星地协同架构，海洋智能浮标、无人机等异构终端通过无线链接入卫星网络。同时，系统采用多核架构的星载边缘服务器实现任务并行处理，每个计算核心具备独立的任务执行能力。通过综合分析任务组的截止期限约束与异构资源分布状态，动态地决策任务的最佳计算位置——本地或卫星边缘服务器。

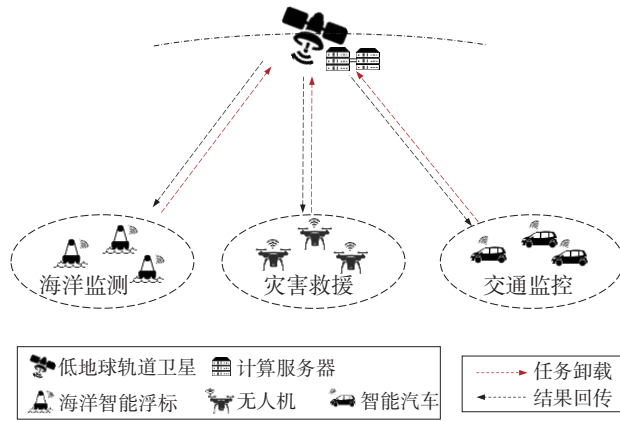


图1 SatEC星地协同架构

1.1 任务模型

该研究采用了参数化建模方法，任务实例可以表述为： $Ta_{\tau,n} = (c_{\tau,n}, d_{\tau,n}, t_{\tau,n})$ ， $(1 \leq n \leq N, 1 \leq \tau \leq T)$ ，其中， $c_{\tau,n}$ 表示任务 $Ta_{\tau,n}$ 所需的CPU单元总量， $d_{\tau,n}$ 表示任务的输入数据量大小， $t_{\tau,n}$ 为任务的最大允许时延，即截止时间。整个系统以时隙模式运行，在每个时隙 τ ($1 \leq \tau \leq T$)中，异构终端会产生一个任务 $Ta_{\tau,n}$ ($1 \leq n \leq N$)，这个任务通过一个二进制决策变量 $x_{\tau,n} \in \{0, 1\}$ 来实现计算节点的动态分配策略。其中， $x_{\tau,n} = 0$ 表示任务在本地终端进行计算， $x_{\tau,n} = 1$ 表示任务卸载给卫星服务器进行计算。

1.2 通信模型

在图1所示框架下，每个地面终端 n 可根据任务需求与LEO卫星提供的无线信道建立连接。每个信道均可为终端动态分配一定的带宽资源，从而确保任务数据的高效、稳定传输。基于Shannon的通信理论模型^[23]，在时间步长 τ 内，地面终端与LEO

卫星之间的数据传输速率 $r_{\tau,n}$ 可由下式得出。

$$r_{\tau,n} = b_{\tau,n} \text{lb} \left(1 + \frac{p_{\tau,n} g_{\tau,n}}{\sum_{n=1}^N x_{\tau,n} p_{\tau,n} g_{\tau,n} + \sigma^2} \right) \quad (1)$$

其中， $b_{\tau,n}$ 为卫星分配给终端 n 的可用带宽资源， $p_{\tau,n}$ 为终端 n 的数据发射功率， $g_{\tau,n}$ 为通信链路的信道增益， $\sum_{n=1}^N x_{\tau,n} p_{\tau,n} g_{\tau,n}$ 计算了所有干扰信号的总功率， σ^2 是加性高斯白噪声（AWGN, additive Gaussian white noise）功率。

1.3 计算模型

1) 本地计算：对于本地执行的任务，处理时延由终端独立计算能力决定，具体表达式为

$$T_{\tau,n}^{\text{local}} = \frac{c_{\tau,n}}{f_{\tau,n}^{\text{local}}} \quad (2)$$

其中， $c_{\tau,n}$ 表示任务 $Ta_{\tau,n}$ 所需的CPU单元总量， $f_{\tau,n}^{\text{local}}$ 代表终端设备的计算速率，定义为单位时间内可执行的CPU周期数。

2) 边缘计算：在卫星边缘计算场景中，任务的处理时延由边缘服务器的单核计算能力和任务本身的计算需求共同决定。具体而言，时延的计算表达式为任务所需的CPU单元总量 $c_{\tau,n}$ 除以边缘服务器的单核计算速率 f^{edge} ，即

$$T_{\tau,n}^{\text{edge}} = \frac{c_{\tau,n}}{f^{\text{edge}}} \quad (3)$$

1.4 时延模型

对于在本地计算的任务来说，不存在传输时延和传播时延，于是任务的本地处理时延即本地的计算时延 $T_{\tau,n}^{\text{local}}$ 。

在任务卸载场景中，终端将待处理任务迁移至边缘服务器执行，带来的时延分别为传输时延、传播时延和计算时延。

1) 传输时延：对于任务 $Ta_{\tau,n}$ ，通信过程包含双向数据交互，首先，需要从终端节点向边缘计算节点发送数据量为 $d_{\tau,n}$ 的原始信息，随后，边缘计算节点须向终端反馈处理结果，但鉴于反馈结果数据规模显著低于输入数据量级，同时LEO卫星下行链路带宽优势明显优于地面通信系统，所以本研究忽略了计算结果回传至地面终端产生的传输时延^[24]。据此，任务 $Ta_{\tau,n}$ 在边缘部署场景中的传输时延可表示为

$$T_{\tau,n}^{\text{trans}} = \frac{d_{\tau,n}}{r_{\tau,n}} \quad (4)$$

2) 传播时延：传播时延主要涉及信号在物理介

质中的传播速度及其需要跨越的距离。在任务卸载至边缘服务器的场景中，这一时延可表述为

$$T_{\tau,n}^{\text{proc}} = \frac{D_{\tau,n}}{c} \quad (5)$$

其中，传播距离 $D_{\tau,n}$ 取决于终端设备 n 与 LEO 卫星之间的实际距离，而信号传播速度 c 接近于光速，本文取固定值 2.97×10^8 m/s。

3) 计算时延：任务的卸载计算时延可从式(3)中获取，即 $T_{\tau,n}^{\text{edge}} = c_{\tau,n}/f^{\text{edge}}$ 。

1.5 任务调度模型

为准确地描述任务处理流程，本文使用时间戳变量 $\Gamma_{\tau,n}$ 来记录任务 $\text{Ta}_{\tau,n}$ 的处理开始时间。对于直接在本地球执行的任務，由于无须等待星载资源，处理开始时间 $\Gamma_{\tau,n}$ 设定为时隙的起始点，即 $\Gamma_{\tau,n} = 0$ 。对于须星载服务器处理的任务，依据传输预处理时间和资源可用性协同决策。可表述为

$$\Gamma_{\tau,n} = \max(T_{\tau,n}^{\text{ready}}, T_{\tau,n}^{\text{idle}}) \quad (6)$$

其中， $T_{\tau,n}^{\text{ready}}$ 表示任务的准备时间，包括从终端到卫星的传输时间及传播时间，即 $T_{\tau,n}^{\text{ready}} = T_{\tau,n}^{\text{trans}} + T_{\tau,n}^{\text{proc}}$ ； $T_{\tau,n}^{\text{idle}}$ 表示星载边缘服务器的实时占用情况：若有空闲服务器，则 $T_{\tau,n}^{\text{idle}}$ 等于 0；若服务器均繁忙，则 $T_{\tau,n}^{\text{idle}}$ 为队列中正在执行任务的最早完成时间。

1.6 问题建模

为系统地分析任务调度机制，通常采用数学建模并运用特定求解策略进行探究。任务 $\text{Ta}_{\tau,n}$ 的全流程时延可定义为

$$T_{\tau,n}^{\text{total}} = (1 - x_{\tau,n})T_{\tau,n}^{\text{local}} + x_{\tau,n}(\Gamma_{\tau,n} + T_{\tau,n}^{\text{edge}} + T_{\tau,n}^{\text{proc}}) \quad (7)$$

该研究的核心目标在于提升任务完成效率，并力求将整体任务的累计时延降至最低。为此，本文将任务调度问题建模为优化问题 P，核心要素为

$$\text{P: } \min \frac{\sum_{n=1}^N T_{\tau,n}^{\text{total}}}{U} \quad (8)$$

s.t.

$$x_{\tau,n} \in \{0, 1\}, \forall n \in (1, 2, \dots, N) \quad (8a)$$

$$T_{\tau,n}^{\text{total}} \leq t_{\tau,n}, \forall n \in (1, 2, \dots, N) \quad (8b)$$

$$\sum_{n=1}^N b_{\tau,n} \leq B_{\text{total}} \quad (8c)$$

$$f_{\min}^{\text{local}} \leq f_{\tau,n}^{\text{local}} \leq f_{\max}^{\text{local}}, \forall n \in (1, 2, \dots, N) \quad (8d)$$

$$p_{\min} \leq p_{\tau,n} \leq p_{\max}, \forall n \in (1, 2, \dots, N) \quad (8e)$$

$$t_{\min} \leq t_{\tau,n} \leq t_{\max}, \forall n \in (1, 2, \dots, N) \quad (8f)$$

其中，系统性能指标 U 代表任务完成率。约束条件中，式(8a)限定任务须在本地球处理或卸载至卫星服

务器执行；式(8b)强制要求任务完成时间不得超出截止期限 $t_{\tau,n}$ ；式(8c)表示系统的传输带宽不能超过最大带宽 B_{total} ；式(8d)与式(8e)分别对物联网终端的计算能力和数据传输功率阈值做出了规范；式(8f)界定了各任务允许的截止时间区间。

2 优化问题求解

由于任务异构性、资源约束和动态环境的复杂耦合，上述优化问题 P 的计算复杂度会随物联网设备规模呈指数增长，这通常被归为 NP-hard 类问题^[25]。传统的优化方法在求解这类问题时存在显著的局限性，例如，基于枚举的精确算法在设备规模扩大时难以在合理时间内收敛^[26]，启发式算法虽能提升效率但无法保证解的质量^[27]。现有 DRL 方法虽然在自适应决策方面展现出潜力，但仍存在单智能体架构难以刻画分层决策特性、时空关联特征提取不足以及奖励函数与任务紧急度映射缺失等问题，导致时敏任务超时率上升与资源分配效率降低。

针对上述挑战，受文献[28]中协同决策框架的启发，本文提出了一种基于自注意力时间卷积网络的多智能体深度强化学习算法来进行问题求解。SATCN-MADRL 算法架构如图 2 所示，该算法框架包含两个功能解耦的智能体：任务调度智能体负责动态评估任务优先级并生成执行序列，资源分配智能体根据实时资源状态决定任务卸载策略。通过构建联合奖励函数和经验共享机制，两个智能体在训练过程中实现策略协同优化，有效地解决了多目标决策中的复杂依赖问题。

2.1 多智能体马尔可夫决策过程

由于无线通信环境与资源的动态特性，整个优化过程本质上是智能体与动态环境的交互过程。针对这种具有可观测性的动态系统，可将优化问题 P 建模为连续状态空间下的多智能体马尔可夫决策过程 (MMDP, multi-agent Markov decision process)。在特定时隙 τ 内，MMDP 可定义为元组 $\langle (s_{\tau}^c, s_{\tau}^o), (a_{\tau}^c, a_{\tau}^o), r_{\tau}, (s_{\tau+1}^c, s_{\tau+1}^o) \rangle$ ，其中，智能体依据实时系统观测状态 $s_{\tau} = (s_{\tau}^c, s_{\tau}^o)$ 选择执行动作 $a_{\tau} = (a_{\tau}^c, a_{\tau}^o)$ ，进而迁移到下一状态 $s_{\tau+1} = (s_{\tau+1}^c, s_{\tau+1}^o)$ 并获取即时奖励 r_{τ} 。通过多轮迭代，智能体将不断优化决策策略。状态空间、动作空间及奖励函数表述如下。

1) 状态空间 S ：多智能体系统的状态空间由各智能体状态的联合空间构成。为明确区分不同智能

体的特征，本文采用变量上标标识机制：以“c”表示任务调度智能体的相关变量，以“o”标记资源分配智能体的对应参数。于是任一时刻 τ 内的状态 $(s_{\tau}^c, s_{\tau}^o) \in S$ 。具体而言，任务调度智能体任一时刻 τ 内的状态 s_{τ}^c 包含多维特征，包括待处理任务的数据规模 $d_{\tau,n}$ 、所需CPU资源总量 $c_{\tau,n}$ 、最大允许时延 $t_{\tau,n}$ 、物联网终端的本地计算能力 $f_{\tau,n}^{local}$ 、数据传输功率配置 $p_{\tau,n}$ ，以及当前通信链路的信道增益 $g_{\tau,n}$ 。可形式化表示为

$$s_{\tau}^c = \{d_{\tau,n}, c_{\tau,n}, t_{\tau,n}, f_{\tau,n}^{local}, p_{\tau,n}, g_{\tau,n}\}, n = (1, 2, \dots, N) \quad (9)$$

资源分配智能体的状态空间构建须与任务调度智能体选定的任务特征动态适配。核心状态参数聚焦于任务处理的时间维度特征：在本地处理场景中，处理过程由起始时间 $\Gamma_{\tau,n}^{local}$ 和执行时间 $T_{\tau,n}^{local}$ 共同表征，其中， $\Gamma_{\tau,n}^{local}$ 在本地场景中固定初始化为0时刻；在卫星服务器处理场景中，起始时间 $\Gamma_{\tau,n}^{edge}$ 须取任务准备就绪时刻与服务器可用时刻的最大值，即前文的式(6)，执行时间 $T_{\tau,n}^{edge}$ 对应卫星服务器的计算时延。由此，该智能体在时隙 τ 内的状态 s_{τ}^o 可形式化定义为两条处理路径的状态元组，即

$$s_{\tau}^o = \{(\Gamma_{\tau,n}^{local}, T_{\tau,n}^{local}), (\Gamma_{\tau,n}^{edge}, T_{\tau,n}^{edge})\} \quad (10)$$

2) 动作空间 A ：与状态空间类似，多智能体系统的动作空间也是由各智能体动作空间的联合构成，任一时刻 τ 内的动作 $(a_{\tau}^c, a_{\tau}^o) \in A$ 。

任务调度智能体的动作空间定义为离散型决策集合，输出动作 a_{τ}^c 表示当前时隙 τ 选择执行的任务索引 $n \in (1, 2, \dots, N)$ ，该决策直接影响任务队列的优先级排序与处理序列。

资源分配智能体的动作空间由混合型决策向量

构成：二元卸载决策 $a_{\tau}^{o1} \in \{0, 1\}$ 确定任务执行位置（0——本地处理，1——卫星卸载）；连续型带宽分配决策 $a_{\tau}^{o2} = b_{\tau,n} \in [0, B_{total}]$ 指定传输带宽资源。二者通过元组形式 $a_{\tau}^o = (a_{\tau}^{o1}, a_{\tau}^{o2})$ 联合表征资源分配策略，共同影响计算资源的时空分布效率。

3) 奖励函数 R ：为了引导智能体在MMDP框架中达成最小化系统总时延与完成率比值的目标，本文将即时奖励 $r_{\tau} \in R$ 直接设计为该目标函数值的负值，即 $r_{\tau} = -(\text{总时延}/\text{完成率})$ 。这样设计是因为强化学习智能体的目标是最大化累积奖励，而当奖励被定义为系统性能指标（总时延/完成率）的负数时，智能体追求最大奖励的行为就等同于自动追求最小化该性能指标，从而完美将系统优化目标转化为智能体的学习目标，可以表述为

$$r_{\tau} = \begin{cases} -\frac{\sum_{n=1}^N T_{\tau,n}^{total}}{U}, & \text{满足约束条件式(8a)~(8f)} \\ -\varphi \frac{\sum_{n=1}^N T_{\tau,n}^{total}}{U}, & \text{其他} \end{cases} \quad (11)$$

其中，当所有约束条件式(8a)~(8f)均被满足时，直接取目标函数的负值；若存在约束违反，则通过一个较大的惩罚系数 φ 对目标函数进行缩放，实施高强度负向激励，从而引导智能体向最小化总时延/完成率这个目标前进。

基于时隙奖励的时序特性，本文通过下式来建立长期奖励计算模型以实现最小化问题到最大化问题的转换。

$$J = \sum_{\tau=1}^T \sum_{n=1}^N \gamma^{\tau} r_{\tau} \quad (12)$$

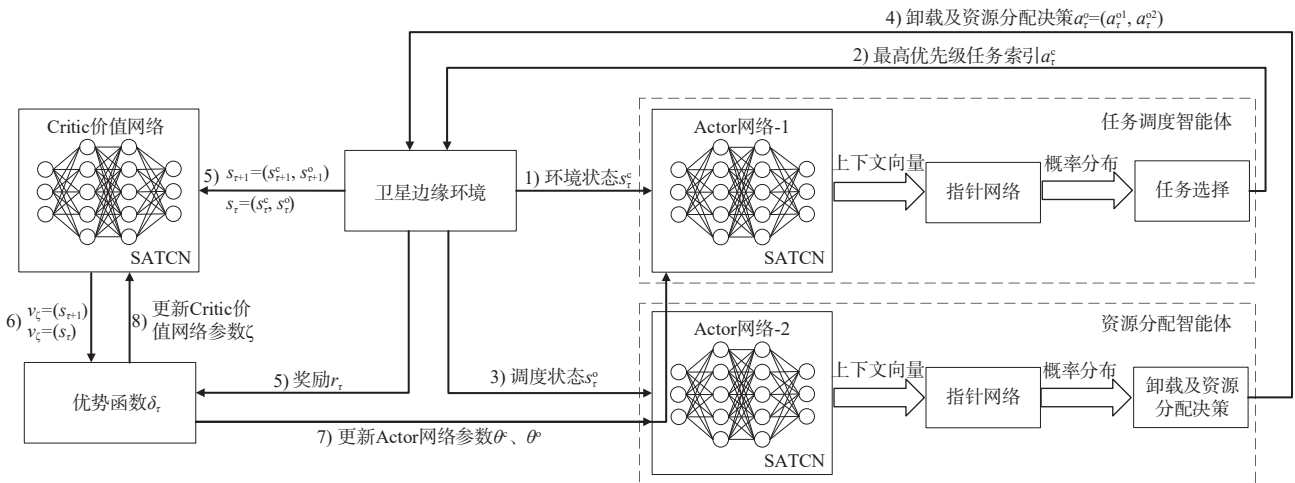


图2 SATCN-MADRL 算法架构

其中，折扣因子 γ 作为强化学习的核心参数，控制着即时奖励与未来奖励的权重分配。

2.2 SATCN-MADRL算法的结构

1) 自注意力时间卷积网络：如图2所示，在SATCN-MADRL框架中，Actor网络-1、Actor网络-2以及Critic价值网络均采用了自注意力时间卷积网络（SATCN, self-attention temporal convolutional network）。传统卷积神经网络（CNN, convolutional neural network）虽在捕获局部时序模式方面表现出色，但固定的感受野限制了其对动态环境中长程依赖的适应能力，而循环神经网络（RNN, recurrent neural network）虽能处理序列数据，却受制于梯度消失和并行计算瓶颈^[29]。为此，本文创新性地设计了SATCN，通过并联自注意力机制^[30]与时间卷积模块^[31]，构建了一个具备多尺度特征感知能力的混合编码器。自注意力时间卷积网络结构如图3所示，该网络采用双通道架构：自注意力分支通过多头注意力机制捕捉全局任务关联，时间卷积分支利用膨胀因果卷积精准提取局部时序特征，最终通过特征融合层实现时空特征的深度协同与高效表达。

具体而言，自注意力模块采用缩放点积注意力机制对输入序列进行变换。首先，SATCN把输入状态输入到全连接网络构成的嵌入操作中生成嵌入向量 $(e_{\tau,1}, e_{\tau,2}, \dots, e_{\tau,N})$ 。接着，自注意力模块基于嵌入向量的各个元素经线性变换分别生成对应的查询向量 $q_{\tau,n} = w_q \cdot e_{\tau,n}$ 、键向量 $k_{\tau,n} = w_k \cdot e_{\tau,n}$ 和值向量 $v_{\tau,n} = w_v \cdot e_{\tau,n}$ 。随后，系统通过计算每个查询向量与全部键向量的点积结果，生成表征元素间关联

强度的注意力权重矩阵。为提升数值稳定性并形成有效概率分布，这些原始权重须经过softmax函数进行标准化处理并得到注意力分数 $\chi_{\tau,n}^j$ ，数学表达式为

$$\chi_{\tau,n}^j = \text{softmax} \left(\frac{q_{\tau,n} \cdot k_{\tau,j}^T}{\sqrt{d_k}} \right), j = (1, 2, \dots, N) \quad (13)$$

其中， d_k 为键向量的维度。

接着，算法将生成的权重系数作用于值向量 $v_{\tau,n}$ 进行线性组合，最终输出各元素的特征表示。这种加权融合机制使每个位置能够动态地聚合序列中其他元素的上下文信息，数学表达式为

$$O_{\tau,n}^1 = \sum_{j=1}^N \chi_{\tau,n}^j \cdot v_{\tau,j}, j = (1, 2, \dots, N) \quad (14)$$

时序卷积网络（TCN, temporal convolutional network）采用的是3层残差因果卷积块，每个模块均包含2个膨胀因果卷积。因果卷积通过约束卷积核作用于历史时刻数据，确保时序关系的严格有序性，膨胀卷积通过指数级扩展采样间隔来增大感受野，从而能够捕捉更长期的时序依赖，数学表征可定义为

$$y_{\tau} = \sum_{k=0}^{K-1} f(k) \cdot x_{\tau-kd} \quad (15)$$

其中， $f(k)$ 是可学习的权重向量， $x_{\tau-kd}$ 表示输入序列在时间点 $\tau - kd$ 的值， K 和 d 分别表示卷积核的大小和膨胀因子。

接着，本文再对TCN网络添加残差连接以防止梯度消失或梯度爆炸，该过程可描述为

$$O_{\tau,n}^2 = \text{ReLU}(\text{Conv}(e_{\tau,n}) + \Omega(e_{\tau,n})) \quad (16)$$

其中， $\Omega(e_{\tau,n})$ 是经过2层膨胀因果卷积后得到的结

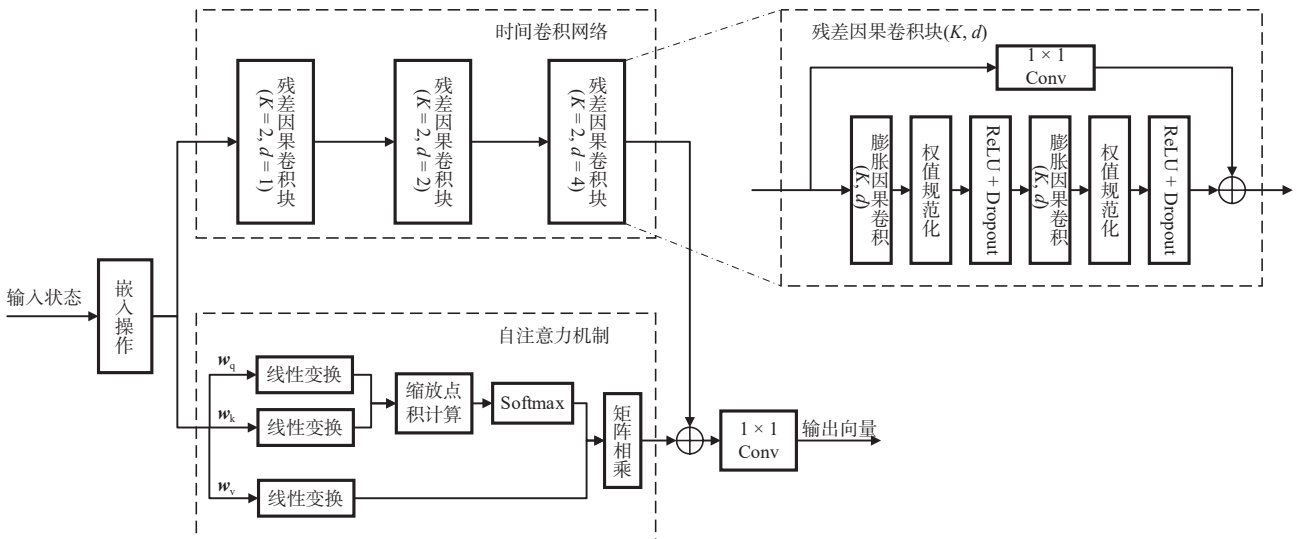


图3 自注意力时间卷积网络结构

果。 $\text{Conv}(e_{\tau,n})$ 是残差连接中的卷积操作，卷积核大小为1，用于维度调整。 $\text{ReLU}(\cdot)$ 为激活函数。

最后，本文将两个分支的输出结果相加，并将它们输入到一个卷积核大小为1的全卷积网络中，从而得到编码器的输出，即上下文向量。这一过程可表示为

$$O_{\tau,n} = \text{Conv}(O_{\tau,n}^1 + O_{\tau,n}^2), n \in (1, 2, \dots, N) \quad (17)$$

2) 指针网络：在动态任务调度场景中，任务的优先级排序需要建立输入与输出序列间的映射关系。为此，本文引入指针网络机制^[32]，通过注意力权重来动态生成任务优先级的概率分布。

首先，该研究对Actor网络的输出 $O_{\tau,n}$ 取平均值 $\bar{O}_\tau = (O_{\tau,1} + O_{\tau,2} + \dots + O_{\tau,N})/N$ ，获得表征全局上下文信息的基准向量 \bar{O}_τ 。在此基础上，分别对基准向量 \bar{O}_τ 和每个时刻的输出状态 $O_{\tau,n}$ 进行线性嵌入操作，生成指针向量 $\mathbf{Pt}_\tau = \mathbf{w}_\tau \cdot \bar{O}_\tau$ 与匹配向量 $\mathbf{Ma}_{\tau,n} = \mathbf{w}_m \cdot O_{\tau,n}$ 集合。

接着，利用缩放点积注意力机制对两者进行相似度计算，同时通过 \tanh 激活函数将注意力得分约束在 $[-C_a, C_a]$ 区间，进而有效地避免了梯度爆炸或数值溢出问题。这种注意力权重本质上构建了任务间的动态关联强度，高权重值直接对应任务执行的高优先级。

最终，使用 softmax 函数将注意力权重归一化为动作选择的概率分布，从而为任务调度提供明确的指导。同时，为满足任务排序的唯一性约束，本文设计了掩码机制：在每次选择任务后，通过将已选任务的匹配向量置为负无穷，经 softmax 函数后概率即变为0，进而显式地消除其后续被重复选择的概率空间。综上所述，最后可得到任务优先级的概率分布，数学表达式为

$$\text{pro}_{\tau,n} = \begin{cases} \text{softmax}(C_a \cdot \tanh(\frac{\mathbf{Pt}_\tau^\top \cdot \mathbf{Ma}_{\tau,n}}{\sqrt{d_m}})), \text{任务}n \text{待调度} \\ 0, \text{任务}n \text{已调度} \end{cases} \quad (18)$$

其中， d_m 为匹配向量 $\mathbf{Ma}_{\tau,n}$ 的维度， \mathbf{Pt}_τ^\top 是指针向量 \mathbf{Pt}_τ 的转置， C_a 为注意力剪辑参数。

3) 任务选择、卸载及资源分配策略：基于任务调度智能体生成的优先级概率分布和资源分配智能体输出的卸载权重矩阵，系统通过分层决策机制实现任务处理策略的动态生成。

对于任务调度智能体，采用自适应掩码增强的随机采样策略，在每个时间步依据当前未被选中的任务概率分布选择执行对象，通过索引映射将高维概率向量转化为具体的物联网终端任务执行序列，该机制既保证了训练阶段对解空间的充分探索，又通过掩码迭代消除了已选任务的概率，进而实现序列唯一性约束。

资源分配智能体通过注意力权重的多维映射，将连续动作空间分解为二进制卸载决策向量和资源分配矩阵，其中，卸载策略通过概率阈值比较确定任务本地执行或边缘卸载，而带宽资源分配依据注意力权重归一化生成资源分配比例系数，实现卫星边缘计算环境中的带宽资源分配。

训练过程采用了带熵正则化的策略梯度方法来平衡探索与利用，通过Critic网络对联合动作的时序价值进行评估，使两个智能体在资源竞争场景中学习到协同博弈策略；测试阶段采用了贪心策略方式，确保模型做出最可能正确的决策。

2.3 SATCN-MADRL算法的训练过程

如图2所示，SATCN-MADRL算法采用双智能体协同架构，包含任务调度与资源分配两个Actor网络，以及共享的Critic价值评估模块。系统初始化阶段主要完成3个核心组件的参数设定：Actor网络-1的参数 θ^c 、Actor网络-2的参数 θ^o 和Critic价值网络的参数 ζ 。在训练迭代过程中，Actor网络-1首先基于环境观测状态 $s_\tau^c = \{d_{\tau,n}, c_{\tau,n}, t_{\tau,n}, f_{\tau,n}^{\text{local}}, p_{\tau,n}, g_{\tau,n}\}$ ， $n = 1, 2, \dots, N$ ，输出上下文变量，接着经指针网络生成概率分布，最后通过动作采样输出调度决策 a_τ^c ，该决策将作为Actor网络-2的输入约束条件。随后，Actor网络-2基于接收到的调度状态 $s_\tau^o = \{(\Gamma_{\tau,n}^{\text{local}}, T_{\tau,n}^{\text{local}}), (\Gamma_{\tau,n}^{\text{edge}}, T_{\tau,n}^{\text{edge}})\}$ 构建任务卸载概率模型，联合输出卸载决策与资源分配方案 $a_\tau^o = (a_\tau^{o1}, a_\tau^{o2})$ 。当双智能体的联合动作 (a_τ^c, a_τ^o) 作用于环境后，系统基于环境反馈的即时奖励 r_τ 和状态转移数据 $s_{\tau+1} = (s_{\tau+1}^c, s_{\tau+1}^o)$ 构建优势函数 $\delta_\tau = r_\tau + \gamma V_\zeta(s_{\tau+1}^c, s_{\tau+1}^o) - V_\zeta(s_\tau^c, s_\tau^o)$ ，通过计算实际收益与Critic价值网络预测值的偏差量，为策略优化提供梯度方向指导：首先，两个Actor网络基于策略梯度算法以最大化优势函数为目标进行参数调整，其梯度可以具体描述为

$$\nabla L_{\theta^c} = -\delta_\tau \nabla \ln \pi_{\theta^c}(a_\tau^c | s_\tau^c) \quad (19)$$

$$\nabla L_{\theta^o} = -\delta_\tau \nabla \ln \pi_{\theta^o}(a_\tau^o | s_\tau^o) \quad (20)$$

接着，Critic 价值网络通过最小化预测值与实际收益的均方误差 $L_c = \delta_t^2 / 2$ 来提升状态价值估计精度，其梯度为

$$\nabla L_c = -\delta_t \nabla V_c(s_t^c, s_t^o) \quad (21)$$

整个训练过程通过迭代循环执行流程，持续优化双智能体策略与价值评估的协同能力，直至达到预设的训练终止条件，最终实现复杂环境中任务调度与资源分配的高效决策。SATCN-MADRL 算法如算法 1 所示。

算法 1 SATCN-MADRL 算法

输入 卫星边缘环境的当前状态 $s_t^c = \{d_{\tau,n}, c_{\tau,n}, t_{\tau,n}, f_{\tau,n}^{local}, p_{\tau,n}, g_{\tau,n}\}$

输出 最合适的调度决策 (a_t^c, a_t^o)

初始化 参数 $\theta^c, \theta^o, \zeta$

for 时隙间隔 $\tau = 1$ **to** 总时隙个数 **do**

将当前状态 s_t^c 输入到 Actor 网络-1 中得到动作概率分布；

采用自适应掩码增强的随机采样策略获取优先任务索引 a_t^c ；

基于 a_t^c 得到调度状态 $s_t^o = \{(I_{\tau,n}^{local}, T_{\tau,n}^{local}), (I_{\tau,n}^{edge}, T_{\tau,n}^{edge})\}$ ；

将 s_t^o 输入 Actor 网络-2 中得到卸载概率分布；

通过概率加权分配机制来实现卸载决策 a_t^o 及资源分配比例 a_t^{o2} ；

动作 (a_t^c, a_t^o) 与环境交互，得到即时奖励 r_t 和下一状态 (s_{t+1}^c, s_{t+1}^o) ；

构建优势函数 $\delta_t = r_t + \gamma V_c(s_{t+1}^c, s_{t+1}^o) - V_c(s_t^c, s_t^o)$ ；

更新网络参数：

$$\theta^c = \theta^c + \alpha_1 \delta_t \nabla \ln \pi_{\theta^c}(a_t^c | s_t^c)$$

$$\theta^o = \theta^o + \alpha_2 \delta_t \nabla \ln \pi_{\theta^o}(a_t^o | s_t^o)$$

$$\zeta = \zeta + \beta \delta_t \nabla V_c(s_t^c, s_t^o)；$$

$$s_t^c, s_t^o \leftarrow s_{t+1}^c, s_{t+1}^o；$$

end for

3 仿真结果与分析

3.1 仿真设置

本文的仿真实验是在一台配备 Intel Core i9-13900K CPU, NVIDIA GeForce RTX 4080 GPU 和 64 GB RAM 的 Windows 10 工作站上进行的。同时使用 Python 3.9 开发了定制化的仿真环境，该环境负责生成异构任务与系统状态，精确计算通信、计算、时延及调度模型，并维护系统运行状态。

SATCN-MADRL 算法中的 Actor 网络、Critic 网络及指针网络使用 PyTorch 2.0.1 实现，并利用 GPU 加速功能进行高效训练。算法的训练流程严格遵循算法 1，策略梯度更新和 Critic 网络更新均在 PyTorch 框架内完成。实验管理、参数配置、结果记录均通过 Python 脚本实现。

实验设定了一个 1 200 m×1 200 m 的动态空域仿真环境，其中，卫星节点遵循预定轨道运动，地面终端设备采用随机分布策略进行传感数据采集。在系统参数配置方面，最大通信带宽 B_{total} 被设定为 40 Mbit/s，加性高斯白噪声功率 σ^2 被设定为 1.0×10^{-11} W，仿真详细参数设置见表 1。

表 1 仿真详细参数设置

参数	数值
物联网终端的数量 N	{10, 20, 30, 40, 50}
多核服务器的核心数 M	{1, 2, 3, 4, 5}
任务的输入数据量大小 $d_{\tau,n}$	400~1 900 KB (均匀分布)
任务所需的 CPU 单元总量 $c_{\tau,n}$	0.2~1.0 Gcycle (均匀分布)
任务的最大允许时延 $t_{\tau,n}$	0.2~0.6 s (均匀分布)
物联网终端的计算能力 $f_{\tau,n}^{local}$	1~5 Gcycle/s (均匀分布)
单核卫星服务器的计算能力 f^{edge}	15 Gcycle/s
最大通信带宽 B_{total}	40 Mbit/s
物联网终端的数据传输功率 $p_{\tau,n}$	1~5 W (均匀分布)
通信链路的信道增益 $g_{\tau,n}$	-40~-20 dB (均匀分布)
加性高斯白噪声功率 σ^2	1.0×10^{-11} W (均匀分布)
物联网终端与卫星间的距离 $D_{\tau,n}$	400~2 000 km (均匀分布)
无线信号的传播速度 c	2.97×10^8 m/s

在算法层面，本文创新性地设计了一种 SATCN 网络架构，该架构融合了注意力机制与时序卷积模块。通过设置卷积核尺寸为 2 并引入 0.2 的神经网络丢弃率 (Dropout 率)，有效地平衡了模型的复杂度与泛化能力。在网络训练过程中，批量规模被设置为 24，隐藏层维度为 128 个神经元。SATCN-MADRL 算法的超参数设置见表 2。

3.2 SATCN-MADRL 算法的性能评估

为验证所提方案的有效性，本文采用任务总时延和任务完成率作为性能指标，并在不同场景中与 5 种现有算法进行对比分析。具体如下。

1) DRTO: 由 Yang 等^[18]通过深度强化学习框架实现的任务调度优化，在降低端到端时延方面具有显著优势；

表2 SATCN-MADRL算法的超参数设置

参数	参数值
Actor网络-1的学习率 α_1	5.0×10^{-6}
Actor网络-2的学习率 α_2	5.0×10^{-6}
Critic价值网络的学习率 β	5.0×10^{-6}
批处理大小	24
隐藏层维度	128
折扣因子 γ	0.99
时间卷积网络的卷积核尺寸 K	2
Dropout率	0.2
注意力剪辑参数 C_a	10

2) BDQN: 由 Sun 和 He^[19]提出的多分支深度Q网络架构, 该算法在动态资源调度场景中展现出良好的任务编排能力;

3) 遗传算法^[13]: 遗传算法作为经典优化技术, 在求解任务分配近似最优解时表现出较强的全局搜索特性;

4) 全卫星执行策略: 将所有计算任务均传输至星载服务器处理;

5) 全本地执行策略: 所有计算任务均在终端设备上自主完成。

首先, 本研究在实验条件设定为20个物联网终端和星载服务器核心数为2的情况下, 比较了不同策略下任务输入数据量对任务总时延和任务完成率的影响。实验结果表明, 数据量的增加会导致总时延的显著上升和任务完成率的下降。在比较不同执行策略时, 全本地执行策略由于受到终端计算能力的限制, 任务输入数据量大小与任务总时延的关系见表3, 任务输入数据量大小与任务完成率的关系如图4所示, 表现出了最高的总时延和较低的任务完成率。全卫星执行策略虽然能够利用星载服务器的高性能计算资源, 但是受到数据传输时延和服务器的资源限制影响, 在大规模任务处理时容易造成队列积压, 同样会产生较高时延及较低的任务完成率。DRTO、BDQN等专注于资源调度的单智能体方法, 状态特征表示难以清晰地分离环境资源状态与任务本身的特性(如紧急度)。这种架构上的局限性导致其难以像本文所提的SATCN-MADRL算法那样有效地感知和响应任务的差异性需求。因此, DRTO和BDQN的性能不如本文所提的SATCN-MADRL算法。例如, 在任务数据量为1 600~1 900 KB时, SATCN-MADRL算法所需的任务总时延较DRTO、

BDQN分别降低了5.4%和4.5%, 任务完成率较DRTO、BDQN分别提高了3.8%和3.0%。SATCN-MADRL算法的这一优异表现得益于多个技术优势: 首先, 基于SATCN特征提取网络的精确任务状态感知, 为调度需求提供了详细的信息; 其次, 多智能体协同机制动态调整任务卸载决策, 有效地平衡了本地与卫星的资源负载; 最后, 通过在线强化学习不断优化策略, 在复杂且动态的环境中实现了时延最小化和任务可靠性的高效协同。

表3 任务输入数据量大小与任务总时延的关系

输入数据量/KB	任务总时延/s					本文算法
	BDQN	DRTO	遗传算法	全卫星执行策略	全本地执行策略	
400~700	1.333 2	1.353 0	1.414 5	2.224 7	2.592 6	1.305 4
700~1 000	2.114 1	2.154 4	2.200 8	3.431 9	3.992 6	2.004 9
1 000~1 300	2.818 3	2.901 8	2.972 6	4.639 0	5.392 6	2.711 4
1 300~1 600	3.705 0	3.754 6	3.827 2	5.846 2	6.792 6	3.592 0
1 600~1 900	4.487 4	4.528 0	4.599 0	7.053 4	8.192 6	4.283 9

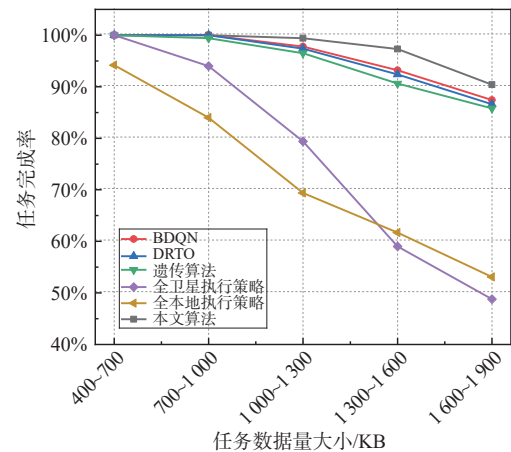


图4 任务输入数据量大小与任务完成率的关系

其次, 本文比较了不同策略下物联网终端数量对任务总时延和任务完成率的影响。实验结果显示, 在星载服务器核心数为2的条件下, 物联网终端数量与系统性能间存在显著关联。物联网终端数量与任务总时延的关系见表4, 随着终端数量从10增加到50, 任务总时延呈现非线性增长趋势。这是由于终端规模扩大, 任务生成量激增, 而有限的星

载计算资源引发任务排队时延呈指数级攀升。进一步地，物联网终端数量与任务完成率的关系如图5所示。在终端规模超过20的场景中，全卫星执行策略的任务完成率骤降至41.4%，根本原因在于星载服务器的资源竞争加剧，导致部分任务错过处理窗口。同时，随着环境中需决策任务数量激增，DRTO、BDQN等单智能体方法因其联合决策空间过于庞大且动态适应能力不足，难以高效实时地响应状态变化并做出最优卸载与资源分配决策。在此背景下，本文所提的SATCN-MADRL算法展现出了显著的性能优势。如表4所示，该算法在终端数量为40的场景中的总时延较传统全卫星执行策略降低了55.6%，而图5的任务完成率在终端数量为40的场景中较传统全本地执行策略提升了17.1%，较DRTO、BDQN等单智能体方法分别提升了3.2%、2.2%，这直接印证了本文所提算法的效能。

表4 物联网终端数量与任务总时延的关系

物联网终端数量	任务总时延/s					
	BDQN	DRTO	遗传算法	全卫星执行策略	全本地执行策略	本文算法
10	1.438 9	1.459 6	1.490 2	1.769 8	3.132 5	1.389 4
20	3.636 8	3.679 1	3.721 1	6.013 7	6.217 1	3.527 6
30	6.351 0	6.440 7	6.531 8	12.688 0	9.610 3	6.276 8
40	9.885 9	10.018 0	10.155 1	21.928 3	13.392 4	9.740 2
50	14.863 4	15.091 0	15.432 2	34.053 6	20.068 4	14.009 3

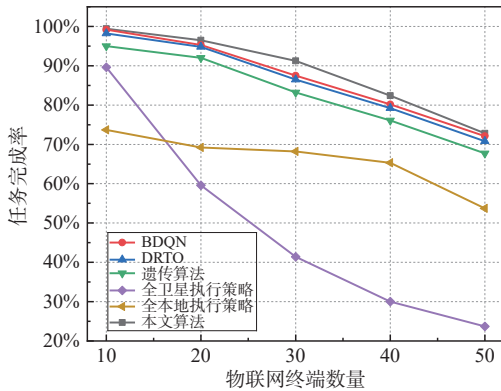


图5 物联网终端数量与任务完成率的关系

接着，本文比较了星载服务器搭载不同数量核心时对任务总时延和任务完成率的影响。实验在固定物联网终端数量为20的场景中展开，星载服务器的核心数与任务总时延的关系见表5，数据显示，随着星载服务器核心数的增加，各算法的总时延呈显著下降趋势。这种改善源于并行计算能力的

提升有效地缓解了任务队列积压，其中，SATCN-MADRL算法的时延降幅尤为明显，显著优于DRTO和BDQN。值得注意的是，全本地执行策略因完全依赖终端算力，时延数据保持稳定，与卫星计算资源配置无关。星载服务器的核心数与任务完成率的关系如图6所示，进一步通过任务完成率指标验证了系统的可靠性变化，当星载服务器核心数从1增加到5时，各算法的任务完成率曲线呈现阶梯式提升。SATCN-MADRL算法的任务完成率始终领先于DRTO和BDQN，尤其在核心数较少（1或2）资源受限的情况下，优势更为突出。这主要归结于SATCN-MADRL算法的多智能体协同机制能够更精细地感知系统资源的动态变化并捕获单个任务的差异性属性（如紧急度），从而在资源紧张时更好地优化调度，避免了传统单智能体（如DRTO、BDQN）仅聚焦于资源分配而忽略任务特性所导致的效率瓶颈。

表5 星载服务器的核心数与任务总时延的关系

星载服务器的核心数	任务总时延/s					
	BDQN	DRTO	遗传算法	全卫星执行策略	全本地执行策略	本文算法
1	4.001 1	4.202 9	4.362 6	10.995 7	6.217 1	3.845 6
2	3.636 8	3.660 5	3.721 1	6.013 7	6.217 1	3.527 6
3	3.278 0	3.328 9	3.368 3	4.373 1	6.217 1	3.149 1
4	3.104 5	3.146 6	3.264 9	3.521 0	6.217 1	2.895 6
5	2.758 4	2.800 2	2.872 0	3.021 9	6.217 1	2.696 8

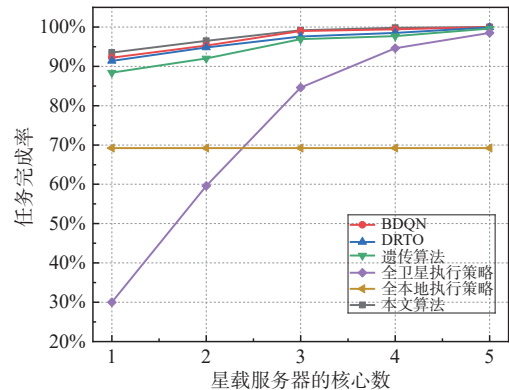


图6 星载服务器的核心数与任务完成率的关系

最后，本文对比了不同任务最大允许时延对系统性能的影响。任务最大允许时延与任务总时延的关系见表6，任务最大允许时延与任务完成率的关系如图7所示，在固定终端规模N = 20和星载服务

器核心数量 $M = 2$ 的条件下，SATCN-MADRL 算法在时延控制和任务完成率方面同样展现出了显著优势。实验数据表明，随着任务最大允许时延要求的放宽，SATCN-MADRL 算法通过智能化的任务调度机制，总时延指标较 BDQN、DRTO 和遗传算法分别降低了 2.7%、4.7% 和 6.9%，同时任务完成率分别实现了 3.9%、4.8% 和 6.8% 的显著提升。这种性能优势源于算法对时间敏感型任务的精准调度能力。相比之下，BDQN 和 DRTO 等专注于资源调度的方法，其单一的策略网络难以将环境资源状态与任务本身的紧急度等关键特征有效地分离建模。这使它们在高时效要求场景中难以像 SATCN-MADRL 算法那样精准地调度不同时间敏感性的任务，容易造成响应滞后或对优先级任务处理不当，进而显著地影响了系统的响应速度和任务完成率。

表6 任务最大允许时延与任务总时延的关系

任务最大允许时延/s	任务总时延/s					本文算法
	BDQN	DRTO	遗传算法	全卫星执行策略	全本地执行策略	
0.2~0.3	4.533 7	4.741 8	4.966 0	6.042 4	9.194 0	4.364 6
0.3~0.4	4.368 0	4.412 5	4.463 5	6.042 4	9.194 0	4.283 1
0.4~0.5	4.253 6	4.280 8	4.359 4	6.042 4	9.194 0	4.123 8
0.5~0.6	4.133 1	4.222 9	4.303 9	6.042 4	9.194 0	4.041 6

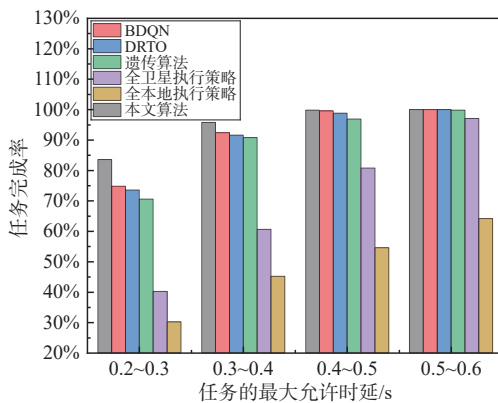


图7 任务最大允许时延与任务完成率的关系

4 结束语

本文针对卫星边缘计算系统中星地资源动态协同与时空敏感任务调度难题，提出了基于 SATCN-MADRL 的联合优化算法。通过构建多智能体分层决策框架，创新性地将任务优先级排序与资源分配进行协同优化，突破了传统单智能体模型在时空关

联特征提取方面的局限性。结合自注意力机制与时间卷积网络的混合神经网络，有效地捕捉了星地协同场景中资源状态与任务特征的时空耦合关系，同时基于概率模型的动态调度机制实现了任务紧急度与资源约束的量化映射。仿真结果表明，相较于 BDQN、DRTO 和遗传算法等基准算法，该算法在任务完成率与时延控制方面均实现了显著的提升，验证了算法在复杂卫星边缘计算场景中的有效性。未来，将着力提升算法在星载计算平台上的部署效率，并将应用场景从单星扩展至卫星星座。为此，计划探索基于图神经网络的动态星间拓扑建模方法，并设计高效的分布式多智能体训练框架。

参考文献：

- [1] Chen H L, Wen Y Q, Huang Y M, et al. Edge computing enabling Internet of ships: a survey on architectures, emerging applications, and challenges[J]. IEEE Internet of Things Journal, 2025, 12(2): 1509-1528.
- [2] Li D B, Sun Y C, Peng J L, et al. Dual network computation offloading based on DRL for satellite-terrestrial integrated networks[J]. IEEE Transactions on Mobile Computing, 2025, 24(3): 2270-2284.
- [3] Zhou J, Zhao Y C, Zhao L, et al. Adaptive task offloading with spatiotemporal load awareness in satellite edge computing[J]. IEEE Transactions on Network Science and Engineering, 2024, 11(6): 5311-5322.
- [4] Zuo Y, Yue M Y, Yang H Y, et al. Integrating communication, sensing and computing in satellite Internet of things: challenges and opportunities[J]. IEEE Wireless Communications, 2024, 31(3): 332-338.
- [5] Zhang C Z, Yang J. An energy-efficient collaborative offloading scheme with heterogeneous tasks for satellite edge computing[J]. IEEE Transactions on Network Science and Engineering, 2024, 11(6): 6396-6407.
- [6] Chen Y, Hu J, Zhao J, et al. Qos-aware computation offloading in LEO satellite edge computing for IoT: a game-theoretical approach[J]. Chinese Journal of Electronics, 2024, 33(4): 875-885.
- [7] Liu L, Mao W P, Li W W, et al. Edge computing offloading strategy for space-air-ground integrated network based on game theory[J]. Computer Networks, 2024, 243: 110331.
- [8] Li J, Shi Y, Dai C, et al. A learning-based stochastic game for energy efficient optimization of UAV trajectory and task offloading in space/aerial edge computing[J]. IEEE Transactions on Vehicular Technology, 2025, 74(6): 9717-9733.
- [9] Xi S Y, Shang B D, Zhang H X, et al. Energy optimization in multisatellite-enabled edge computing systems[J]. IEEE Internet of Things Journal, 2024, 11(12): 21715-21726.
- [10] Sun J C, Chen X, Jiang C X, et al. Distributionally robust opti-

- zation of on-orbit resource scheduling for remote sensing in space-air-ground integrated 6G networks[J]. IEEE Journal on Selected Areas in Communications, 2025, 43(1): 382-395.
- [11] Zhou J, Yang Q, Zhao L, et al. Mobility-aware computation offloading in satellite edge computing networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(10): 9135-9149.
- [12] Shuai J Q, Cui H X, He Y J, et al. Dynamic satellite edge computing offloading algorithm based on distributed deep learning[J]. IEEE Internet of Things Journal, 2024, 11(16): 27790-27802.
- [13] Liu Z G, Dong X Q, Wang L, et al. Satellite network task deployment method based on SDN and ICN[J]. Sensors, 2022, 22(14): 5493.
- [14] Zhang Z Y, Dong T, Yin J, et al. A particle swarm optimization-based queue scheduling and optimization mechanism for large-scale low-earth-orbit satellite communication networks[J]. Sensors, 2025, 25(4): 1069.
- [15] Zhuo M, Feng Y M, Yang P, et al. Optimizing topology in satellite-UAV collaborative IoT: a graph partitioning simulated annealing approach[J]. Drones, 2024, 8(2): 44.
- [16] Zhang S, Cai T Z, Wu D, et al. IoRT data collection with LEO satellite-assisted and cache-enabled UAV: a deep reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2024, 73(4): 5872-5884.
- [17] Waqar N, Ali Hassan S, Mahmood A, et al. Computation offloading and resource allocation in MEC-enabled integrated aerial-terrestrial vehicular networks: a reinforcement learning approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(11): 21478-21491.
- [18] Yang W T, Liu Z B, Liu X W, et al. Deep reinforcement learning-based low-latency task offloading for mobile-edge computing networks[J]. Applied Soft Computing, 2024, 166: 112164.
- [19] Sun Y, He Q J. Joint task offloading and resource allocation for multi-user and multi-server MEC networks: a deep reinforcement learning approach with multi-branch architecture[J]. Engineering Applications of Artificial Intelligence, 2023, 126: 106790.
- [20] Chai F R, Zhang Q, Yao H P, et al. Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT[J]. IEEE Transactions on Vehicular Technology, 2023, 72(6): 7783-7795.
- [21] Zhang H Y, Liu R K, Kaushik A, et al. Satellite edge computing with collaborative computation offloading: an intelligent deep deterministic policy gradient approach[J]. IEEE Internet of Things Journal, 2023, 10(10): 9092-9107.
- [22] Cai H N, Yang X M, Wu H N, et al. Freshness-aware task offloading and resource scheduling for satellite edge computing[C]//Proceedings of the 2024 IEEE Wireless Communications and Networking Conference (WCNC). Piscataway: IEEE Press, 2024: 1-6.
- [23] Shannon C E. A mathematical theory of communication[J]. Bell System Technical Journal, 1948, 27(3): 379-423.
- [24] Song Z Y, Hao Y Y, Liu Y W, et al. Energy-efficient multiaccess edge computing for terrestrial-satellite Internet of things[J]. IEEE Internet of Things Journal, 2021, 8(18): 14202-14218.
- [25] Cheng L, Feng G, Sun Y, et al. Energy-constrained satellite edge computing for satellite-terrestrial integrated networks[J]. IEEE Transactions on Vehicular Technology, 2025, 74(2): 3359-3374.
- [26] Lyu T, Xu Y Q, Liu F F, et al. Task offloading and resource allocation for satellite-terrestrial integrated networks[J]. IEEE Internet of Things Journal, 2025, 12(1): 262-275.
- [27] Qu Y B, Zhang T T, Feng Y, et al. Computation offloading and resource allocation for E2E tasks in satellite edge computing networks[J]. Space: Science & Technology, 2024, 4: 144.
- [28] Xiong J Y, Guo P, Wang Y, et al. Multi-agent deep reinforcement learning for task offloading in group distributed manufacturing systems[J]. Engineering Applications of Artificial Intelligence, 2023, 118: 105710.
- [29] Mienye I D, Swart T G, Obaido G. Recurrent neural networks: a comprehensive review of architectures, variants, and applications[J]. Information, 2024, 15(9): 517.
- [30] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30: 5998-6008.
- [31] Bai S, Kolter J Z, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[PP]. arXiv preprint, 2018, arxiv: 1803.01271.
- [32] Vinyals O, Fortunato M, Jaitly N. Pointer networks[C]//Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015). Montreal, Canada: Curran Associates, Inc., 2015: 2692-2700.

[作者简介]



陈娟(1986-), 女, 博士, 西华大学计算机与软件工程学院讲师, 主要研究方向为智能边缘计算、深度强化学习。



钟杰(1999-), 男, 西华大学计算机与软件工程学院硕士生, 主要研究方向为卫星边缘计算、移动边缘计算。



吴宗玲(1987-), 男, 博士, 西南交通大学信息科学与技术学院工程师, 主要研究方向为嵌入式物联网。



田谛(2001-), 男, 西华大学计算机与软件工程学院硕士生, 主要研究方向为空天地一体化网络。



陈玉杰(2000-), 男, 西华大学计算机与软件工程学院硕士生, 主要研究方向为移动边缘计算、云计算。