

空地融合车载网络场景下基于GAT-MADRL的缓存策略

朱思峰¹, 史科选¹, 赵伟丰^{1*}, 张宗辉¹, 郁有建¹, 张青华², 朱海³

(1. 天津城建大学 计算机与信息工程学院, 天津 300384; 2. 天津城建大学 图书馆, 天津 300384; 3. 河南工程学院 计算机学院, 郑州, 河南 451191)

摘要: 针对空地融合车载网络中节点异构性强、拓扑动态变化频繁以及协同缓存困难等挑战, 本文构建了面向空地融合车载网络的有向异构图模型, 设计了融合图注意力网络与多智能体深度强化学习的分布式缓存决策框架, 提出了一种基于图注意力网络与多智能体深度强化学习的缓存策略优化方法。实验结果表明, 与随机策略、GA策略、MADQN策略、MADDPG策略相比, 所提的策略在任务平均时延方面分别降低了51.61%、38.12%、26.83%、22.12%, 系统总能耗分别降低了71.68%、67.53%、43.23%、24.21%, 缓存命中率分别提升了261.19%、49.11%、21.43%、6.21%。

关键词: 空地融合车载网络; 缓存策略; 图注意力网络; 多智能体深度强化学习

中图分类号: TP929.5

文献标志码: A

doi: 10.11959/j.issn.2096-3750.XXXX.

Caching Strategy Based on GAT-MADRL for Space-Air-Ground Integrated Vehicular Networks

ZHU Sifeng¹, SHI Kexuan¹, ZHAO Weifeng^{1*}, ZHANG Zonghui¹, YU Youjian¹, ZHANG Qinghua²,
ZHU Hai³

1. School of Computer and Information Engineering, Tianjin Chengjian University, Tianjin, 300384, China

2. Library, Tianjin Chengjian University, Tianjin, 300384, China

3. School of Computer, Henan University of Engineering, Zhengzhou, Henan 451191, China

Abstract: To address the challenges of strong node heterogeneity, frequent dynamic topology changes, and difficult cache coordination in Space-Air-Ground Integrated Vehicular Network (SAGVN), this paper constructs a directed heterogeneous graph model for SAGVNs, designs a distributed caching decision framework that integrates Graph Attention Network (GAT) and Multi-Agent Deep Reinforcement Learning (MADRL), and proposes a caching strategy optimization method based on GAT and MADRL. Experimental results show that, compared to the Random strategy, GA strategy, MADQN strategy, and MADDPG strategy, the proposed strategy reduces the average task delay by 51.61%, 38.12%, 26.83%, and 22.12%, lowers the total system energy consumption by 71.68%, 67.53%, 43.23%, and 24.21%, and improves the cache hit rate by 261.19%, 49.11%, 21.43%, and 6.21%, respectively.

Key words: Space-Air-Ground Integrated Vehicular Network, caching strategy, Graph Attention Network, Multi-Agent Deep Reinforcement Learning

收稿日期: XXXX-XX-XX; 修回日期: XXXX-XX-XX

通信作者: 赵伟丰, zhaowf@tcu.edu.cn

基金项目: 国家自然科学基金项目 (62172457), 天津城建大学研究生创新项目 (2025YKC077, 2025YKC079), 河南省科技发展计划项目 (252103810011, 252103810206, 261111211200)

Foundation Items: National Natural Science Foundation of China (62172457), Postgraduate Innovation Project of Tianjin Chengjian University (2025YKC077, 2025YKC079), Henan Provincial Science and Technology Development Program (252103810011, 252103810206, 261111211200).

0 引言

随着车联网 (IoV, internet of vehicles) 技术的发展, 车载应用对网络覆盖广度与服务响应速度的需求

越来越高^[1-3]。空天地融合车载网络 (SAGVN, space-air-ground integrated vehicular network) 凭借卫星、无人机、地面节点的多层级架构, 成为解决地面网络覆盖盲区、应对动态业务需求的核心技术方向^[4-6]。在该网络架构中, 内容缓存作为降低核心网回传负载、缩短用户请求延迟的关键手段^[7-8], 其性能直接决定了网络服务质量。

然而, SAGVN 的多层级架构^[10], 为分布式缓存策略的设计带来了区别于传统地面网络的根本性挑战。SAGVN 包含低轨卫星 (LEO, low earth orbiting satellite)、无人机 (UAV, unmanned aerial vehicle)、路边单元 (RSU, roadside unit) 及车载终端等多类型节点^[10-11], 不同节点在缓存容量、能源供给、通信链路质量与服务范围上存在本质差异。传统缓存方案大多采用统一决策规则^[12-14], 未考虑节点异构性导致的资源约束差异。

其次, 网络的拓扑动态性破坏了缓存内容的时空关联性。SAGVN 中, UAV 的灵活部署、车载终端的高速移动及 LEO 卫星的轨道运动, 导致节点间链路连通性呈现高动态特征^[15-16]。链路的频繁建立与断开使得缓存节点与用户请求的空间关联持续变化; 链路质量的时变特性, 会导致缓存内容的传输效率波动, 传统静态缓存策略无法实时调整内容的缓存位置与优先级, 进而引发预缓存内容失效、关键内容传输延迟超限等问题^[17-18]。

针对上述挑战, 本文构建了有向异构图实现 SAGVN 的结构化建模, 提出了图注意力网络 (GAT, graph attention network) 与多智能体强化学习 (MADRL, multi-agent deep reinforcement learning) 融合的 GAT-MADRL 框架, 利用 GAT 的注意力机制, 对节点邻居及关联边的重要性进行自适应加权, 最后基于 MADRL 构建低开销的隐式协同决策机制, 达成全局缓存性能最优的目标。

1 相关工作

早期 IoV 缓存研究聚焦地面静态场景, 以固定规则或单一智能算法为核心, 通过局部优化提升缓

存效率, 但难以适配车辆移动与请求。文献[19]为应对网络拓扑频繁变化的挑战, 将复杂的延迟优化问题转化为两个子问题, 并设计了贪心算法进行求解。文献[20]构建了宏基站、路侧单元和云的缓存架构, 设计了社交感知辅助边缘协同缓存算法, 根据用户社交语义相似度和网络状态动态做出智能缓存决策。

随着 SAGVN 的提出, 研究逐渐向多层级缓存拓展, 但现有方案仍存在层级割裂与优化目标单一的问题, 未能实现全局缓存性能最优。文献[21]设计了一个缓存辅助的空天地一体化网络移动边缘计算系统, 来进行四种不同的任务卸载流程。文献[22]提出了一个面向 SAGVN 的联合优化问题框架, 采用了交替优化框架, 将原问题分解为子问题, 并设计了两种多对多交换匹配算法, 从而获得了系统的整体优化解。

值得注意的是, 已有部分研究意识到图网络与强化学习的互补性, 将其用于空天地网络的优化, 但未针对 SAGVN 缓存的核心需求设计融合策略。文献[23]将无线边缘缓存技术引入 SAGVN, 利用中继节点作为缓存节点, 通过预缓存热点内容, 构建了一个缓存增强型 SAGVN 架构, 减少冗余数据传输。文献[24]针对 LEO 卫星网络的高动态特性, 提出了分布式路由算法, 利用图注意力网络对单跳邻居信息进行编码, 捕获了网络的多跳拓扑状态, 降低了通信开销。

综上, 现有研究围绕 IoV 与 SAGVN 缓存优化已形成多维度探索, 智能体由于只针对局部状态而存在协同盲区, 而 GAT 通过图结构将车联网中所有节点的关联关系纳入统一建模, 智能体在决策时可通过 GAT 提取邻居节点状态与全局供需关联的关键信息。针对上述挑战, 本文提出 GAT-MADRL 融合框架: 通过有向异构图建模, 利用 GAT 动态加权节点与链路重要性, 弥补传统静态建模的适配缺陷, 来实现全局性能优化。

2 系统模型

本节针对 SAGVN 的异构性、动态性及缓存协同需求, 构建空间层、空中层、地面层与虚拟层的系统模型, 该模型由 LEO、UAV、RSU 和智能车辆等节点组成。SAGVN 系统架构图如图 1 所示。

假设架构中的车辆位置服从泊松分布, 并且每

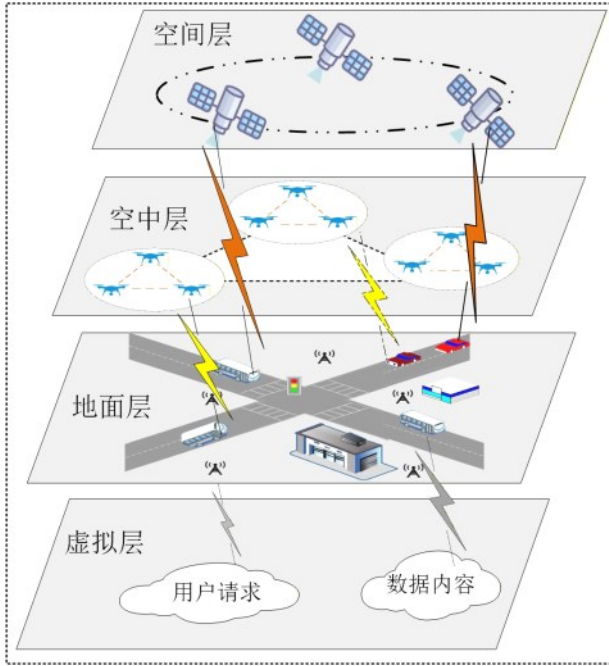


图1 系统模型

辆智能车辆每次仅发起一项内容服务请求，且需在当前请求内容完全交付后，方可发起后续内容请求。

2.1 网络实体建模

本文采用有向异构图 $G = (V, E, R)$ 对 SAGVN 网络实体及交互关系进行结构化建模，其中 $V = \{v_1, v_2, \dots, v_n\}$ 为节点集合，包含物理与虚拟节点， $E = \{e_{1m}, e_{2(m-1)}, \dots, e_{m1}\}$ 为边集合，包含物理链路与逻辑关系， $R = \{r_1, r_2, \dots, r_z\}$ 为边类型集合。

2.1.1 节点定义

节点按物理实体与虚拟逻辑分类，具体节点定义见表1。本文将各类节点统一映射为定长特征向量输入 GAT。对任意节点 i ，定义原始输入为 $x_i = [type_i, a_{i1}, a_{i2}, a_{i3}, a_{i4}]$ ，其中 $type_i$ 为节点类型编码，对应 LEO 卫星、UAV、RSU、车载终端、用户请求和数据内容。对于物理节点， $a_{i1} \sim a_{i4}$ 依次表示剩余缓存容量、剩余计算资源、剩余能量和请求覆盖度；对于用户请求节点， $a_{i1} \sim a_{i4}$ 依次表示任务数据量、计算需求周期数、优先级系数和最大容忍时延；对于数据内容节点， $a_{i1} \sim a_{i4}$ 依次表示内容流行度、内容大小、剩余有效期和副本数；不存在的属性为0。连续特征均采用 min-max 归一化处理。

2.1.2 边与边类型定义

边类型定义如表2所示。对任意有向边 e_{ij} 定义

表1 节点定义

| 节点层级 | 节点类型 |
|------|--------|
| 空间层 | LEO 卫星 |
| 空中层 | UAV |
| 地面层 | RSU |
| 地面层 | 车载终端 |
| 虚拟层 | 用户请求 |
| 虚拟层 | 数据内容 |

6 维边特征向量 $z_{ij} = [r_{ij}, q_{ij}, b_{ij}, d_{ij}, \tau_{ij}, s_{ij}]$ ，其中 r_{ij} 为边类型编码， q_{ij} 表示归一化链路质量， b_{ij} 表示可用带宽， d_{ij} 表示归一化节点间距离， τ_{ij} 表示当前传输时延， s_{ij} 表示链路稳定度。对于逻辑边“物理节点→用户请求”，分别用请求关联强度与服务可达性替代不可观测的物理链路量；对于“物理节点→数据内容”边，用缓存占用状态与内容命中关系表征。

表2 边类型定义

| 边类型 r | 源节点→目标节点 |
|---------|---------------------|
| 星间链路 | LEO 卫星→LEO 卫星 |
| 星地链路 | LEO 卫星→UAV/RSU/车载终端 |
| 无人间链路 | UAV→UAV |
| 空地链路 | UAV→RSU / 车载终端 |
| 地面链路 | RSU→RSU / 车载终端； |
| 车车链路 | 车载→车载 |
| 缓存-内容链路 | UAV/RSU→用户请求 |
| | 物理节点→数据内容 |

2.2 任务模型

针对 SAGVN 中的智能车辆对数据内容的获取需求，引入子任务拆分机制，设所有类型子任务集合为 $Task_{ALL}$ ，车载终端发起的缓存任务 k 统一表示为 $Subtask_k = \{Task_{SET}, T_k^{max}, \rho_k\}$ ，其中 T_k^{max} 为任务 k 的最大容忍延迟， ρ_k 为任务 k 的全局优先级权重，用于子任务资源分配的优先级排序， $Task_{SET}$ 与 $Task_{ALL}$ 的关系表示为

$$Task_{SET} \subseteq Task_{ALL} \quad (1)$$

按缓存可行性将子任务类型分为两类：可缓存型子任务 ST_Y 、不可缓存型子任务 ST_N 。每个子任务统一表征为 ST_* ，如式 (2) 所示。

$$ST_* = \begin{cases} \{T_z, T_D, T_L, D_i, T_{val}, \rho_*\}, & * = Y \\ \{T_z, -, T_L, D_i, -, \rho_*\}, & * = N \end{cases} \quad (2)$$

其中， T_z 为 CPU 周期数， T_D 为所需的内存大小，仅 ST_Y 需定义， T_L 为子任务对应的程序代码文件大小， D_i 为子任务被请求的次数， T_{val} 为有效时长，

仅 ST_Y 需定义。 ρ_* 为优先级系数。 ST_Y 可以存储到节点 n 触发条件如式 (3) 所示。

$$\begin{cases} D_n(t) \geq T_D(ST_Y) \wedge \rho_*(ST_Y) \geq \rho_{\min}(n) \\ D_i(ST_Y) \geq \Gamma_{acc}(Type(n)) \\ T_{val}(ST_Y) \geq T_{val}(n) \end{cases} \quad (3)$$

其中 $D_n(t)$ 为节点 n 在时刻 t 的剩余缓存容量, $\rho_{\min}(n)$ 为节点 n 当前已缓存任务的最低优先级, $\Gamma_{acc}(Type(n))$ 为节点对应的访问频次阈值, $T_{val}(n)$ 为节点 n 在当前区域的覆盖时长。当 $D_n(t) < T_D(ST_Y)$ 时, 按 ρ_*/T_{val} 排序, 淘汰比值最小的已缓存的 ST_Y , 确保缓存资源向高价值、长有效期内容倾斜。

2.3 网络与传输模型

在 GAT-MADRL 框架中卫星、无人机、边缘服务器、车辆各节点构成一个异构通信网络。数据流根据其优先级、带宽需求及节点状态, 在卫星链路和无线链路间进行动态适配与路由, 以确保传输效率^[25-26]。

本文依据香农定理计算卫星、无人机、边缘服务器及智能汽车用户终端之间无线通信链路的数据传输速率 DTR_{ij}^{wt} , 如式 (4) 所示。

$$DTR_{ij} = Blb \left(1 + \frac{p_i H_{ij}}{w_i} \right) \quad (4)$$

其中 i 和 j 分别对应发送设备与接收设备, DTR_{ij} 表示传输速率, B 为信道带宽, P_i 为发射功率, w_i 为噪声功率, H_{ij} 为信道增益。 H_{ij} 如式 (5) 所示

$$H_{ij} = d_{ij}^{-\alpha} \quad (5)$$

其中 α 为路径衰减因子, d_{ij} 表示设备 i 与 j 之间的距离, 如式 (6) 所示。

$$d_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right)^{1/2} \quad (6)$$

其中, x 、 y 、 z 代表设备在三维坐标系中的坐标。

ST_N 任务需上传程序, 传输时延如式 (7) 所示。

$$T_{trans,N}(ST_N) = \frac{T_L(ST_N)}{DTR_{ij}} \quad (7)$$

ST_Y 任务缓存未命中时传输时延如式 (8) 所示。

$$T_{trans,Y}(ST_Y) = \frac{T_L(ST_Y) + T_D(ST_Y)}{DTR_{ij}} \quad (8)$$

计算任务 k 的总时延为其所有子任务传输时延之和, 如式 (9) 所示。

$$T_{trans}(Task_{SET}) = \sum_{ST_N \in Task_{SET}} T_{trans}(ST_N) + \sum_{ST_Y \in Task_{SET}} T_{trans}(ST_Y) \quad (9)$$

2.4 计算模型

2.4.1 节点计算能力建模

SAGVN 各物理节点计算能力以单位时间可提供的最大计算周期数表征记为 F_n 。

当节点 $n \in$ 车载终端时, 车载终端的计算能力 F_{veh} , 如式 (10) 所示。

$$F_{veh} = F_{obu} \cdot \min \left(1, \frac{T_k^{\max}}{T_{comp}(ST_*)} \right) \quad (10)$$

其中, F_{obu} 为本地计算速率, $T_{comp}(ST_*)$ 为计算时长。

当 $n \in$ RSU 时, RSU 的计算能力 F_{rsu} , 如式 (11) 所示。

$$F_{rsu} = F_{edge} \quad (11)$$

其中, F_{edge} 为边缘服务器的计算速率。

当 $n \in$ UAV 时, UAV 的计算能力 F_{uav} , 如式 (12) 所示。

$$F_{uav}(t) = F_{uav}^{\max} \cdot \eta_{uav}(t) \quad (12)$$

其中, F_{uav}^{\max} 为 UAV 的最大速率, $\eta_{uav}(t) \in (0,1]$ 为 t 时刻能量状态系数。

当 $n \in$ LEO 时, LEO 卫星的计算能力 F_{leo} , 如式 (13) 所示。

$$F_{leo} = F_{leo}^{\max} \quad (13)$$

其中, F_{leo}^{\max} 为 LEO 的最大计算周期速率。

进一步定义节点 n 在 t 时刻的剩余计算资源, 如式 (14)。

$$F_n^{res}(t) = F_n(t) - \sum_{ST_k \in \Phi_n(t)} f_{n,k}(t) \quad (14)$$

其中 $\Phi_n(t)$ 为节点正在处理的任务集合; $f_{n,k}(t)$ 为计算资源, 满足 $0 < f_{n,k}(t) \leq F_n^{res}(t)$ 。

2.4.2 任务计算需求建模

子任务的计算需求以完成该任务所需的总计算周期数量化记为 $Cycle_*$, 其与任务数据量、计算复杂度直接相关, 如式 (15) 所示。

$$Cycle_* = \lambda_* \cdot T_z \cdot \rho_* \quad (15)$$

其中, T_z 为所需 CPU 周期数, λ_k 为计算复杂度系数。 ρ_* 为优先级系数。

2.4.3 计算延迟建模

缓存机制作为优化延迟的关键环节, 其命中情况将显著影响任务数据的分发方式: 若缓存未命

中，则需计入从远端源节点获取数据的额外传输延迟。

当子任务在发起节点本地处理，若节点 n 缓存了子任务所需的程序代码与输入数据，则无需数据传输，延迟为计算时延，如式 (16) 所示。

$$T_{comp}^{local}(ST_Y) = \frac{Cycle_*}{f_{n,Y}(t)} \quad (16)$$

若子任务在发起节点本地处理，且缓存未命中，先从关联节点获取缓存内容再进行本地计算，总延迟如式 (17) 所示。

$$T_{comp}^{local}(ST_*) = T_{trans}(ST_*) + \frac{Cycle_*}{f_{n,*}(t)} \quad (17)$$

若子任务从发起节点 n 卸载至目标节点 m 处理，卸载延迟包含三部分：子任务数据传输至目标节点的延迟、目标节点的计算延迟、计算结果回传延迟，如式 (18) 所示。

$$T_{comp}^{offload}(ST_*) = T_{trans}^{off}(ST_*) + \frac{Cycle_*}{f_{m,*}(t)} + T_{trans}^{res}(ST_*) \quad (18)$$

其中， $T_{trans}^{off}(ST_*)$ 为数据上传延迟， $T_{trans}^{res}(ST_*)$ 为计算结果回传延迟， $f_{m,*}(t)$ 为分配给子任务的计算资源，需满足 $f_{m,*}(t) \leq F_m^{res}(t)$ 。

2.4.4 计算能耗建模

能耗是 UAV、车载终端等能源受限节点的核心约束，如式 (19) 所示。

$$E_{comp}(n, ST_*) = \kappa_n \cdot f_{n,*}(t)^2 \cdot Cycle_* \quad (19)$$

其中， κ_n 为计算能耗系数， $f_{n,*}(t)$ 为计算资源， $Cycle_*$ 为总计算周期数。

对于 UAV，计算能耗需纳入飞行能耗、通信能耗预算，如式 (20) 所示。

$$E_{comp}(UAV, ST_*) + E_{flight}(UAV, t) + E_{comm}(UAV, t) \leq E_{res}(UAV, t) \quad (20)$$

其中， $E_{flight}(UAV, t)$ 为 UAV 飞行能耗， $E_{res}(UAV, t)$ 为 t 时刻 UAV 的剩余能量， $E_{comm}(UAV, t)$ 为通信能耗，如式 (21) 所示。

$$E_{comm}(UAV, t) = P_{trans} \cdot T_{trans}(ST_*) \quad (21)$$

其中， P_{trans} 为传输功率。

对于车载终端能耗约束，如式 (22) 所示。

$$E_{comp}(Veh, ST_*) + E_{comm}(Veh, t) \leq E_{bat}(Veh, t) \quad (22)$$

其中， $E_{bat}(Veh, t)$ 为剩余容量。

2.4.5 总延迟与总能耗建模

缓存命中时可省去数据传输环节，显著降低总开销。结合缓存命中标识 $\delta_*(n) \in \{0, 1\}$ ，当

$\delta_*(n) = 1$ 时表示节点 n 缓存任务所需内容，当 $\delta_*(n) = 0$ 时表示节点 n 未缓存该内容。子任务在节点 n 的延迟与能耗可综合表示为式 (23) 和式 (24)。

$$T(ST_*, n) = \frac{Cycle_*}{f_{n,*}(t)} + (1 - \delta_*(n)) \cdot T_{trans}(ST_*, n) \quad (23)$$

$$E(ST_*, n) = \kappa_n \cdot f_{n,*}(t)^2 \cdot Cycle_* + (1 - \delta_*(n)) \cdot E_{trans}(ST_*, n) \quad (24)$$

由式 (23) 和式 (24) 可见，缓存命中不会改变任务在节点上的计算开销，仅消除未命中场景下额外产生的传输时延与传输能耗。其中， $E_{trans}(ST_*, n)$ 为数据传输能耗，其计算如式 (25) 所示。

$$E_{trans}(ST_*, n) = P_{trans} \cdot T_{trans}(ST_*, n) \quad (25)$$

任务 k 的总延迟为所有子任务延迟之和，任务 k 的总能耗为所有子任务能耗之和，如式 (26) 和式 (27) 所示。

$$T(k, n) = \sum_{ST_* \in Task_k} T(ST_*, n) = \sum_{ST_Y \in Task_k} T(ST_Y, n) + \sum_{ST_N \in Task_k} T(ST_N, n) \quad (26)$$

$$E(k, n) = \sum_{ST_* \in Task_k} E(ST_*, n) = \sum_{ST_Y \in Task_k} E(ST_Y, n) + \sum_{ST_N \in Task_k} E(ST_N, n) \quad (27)$$

3 评价标准与问题优化模型

3.1 时延

任务时延包含数据传输时延、计算时延与缓存命中等待时延，如式 (28) 所示。

$$J_1 = \frac{1}{K} \sum_{k=1}^K T(k) \quad (28)$$

其中， $T(k)$ 为任务 k 的总时延。

3.2 全局缓存命中率

缓存命中率直接反映缓存资源的利用效率，如式 (29) 所示。

$$J_2 = \frac{\sum_{k=1}^K \sum_{ST_Y \in Task_k} \delta_*(n)}{\sum_{k=1}^K |Task_k^{ST_Y}|} \quad (29)$$

其中， $|Task_k^{ST_Y}|$ 为任务 k 中可缓存子任务的数量。

3.3 能耗

针对 UAV、车载终端等能源受限节点，通过优化缓存内容的存储与传输策略，减少无效能耗。

$$J_3 = \sum_{n \in n_{ALL}} [E_{comp}(n) + E_{comm}(n) + E_{cache}(n)] \quad (30)$$

其中, n_{ALL} 为所有物理节点集合; $E_{comp}(n)$ 为节点的计算能力; $E_{comm}(n)$ 为节点的通信传输能耗; $E_{cache}(n)$ 为节点的缓存存储能耗, 与缓存内容容量、存储时长正相关。

3.4 问题优化模型

由于不同目标函数的量纲与取值范围存在显著差异, 直接加权求和会导致优化方向偏置。为此, 本文采用 min-max 归一化方法 对各目标函数进行统一化处理, 将其映射至相同的[0,1]区间, 以确保多目标优化结果的合理性与公平性。

$$\tilde{J}_1 = \frac{J_1 - J_1^{\min}}{J_1^{\max} - J_1^{\min}} \quad (31)$$

$$\tilde{J}_2 = J_2 \quad (32)$$

$$\tilde{J}_3 = \frac{J_3 - J_3^{\min}}{J_3^{\max} - J_3^{\min}} \quad (33)$$

其中, J_1^{\min} 、 J_1^{\max} 、 J_3^{\min} 、 J_3^{\max} 由实验统计范围确定。

由于各目标存在耦合关系, 考虑到卸载决策、通信时延、服务器能耗、缓存策略, SAGVN 中的车辆任务卸载问题可以建模为一个联合优化的问题。

$$\begin{cases} \min J = \omega_1 \tilde{J}_1 - \omega_2 \tilde{J}_2 + \omega_3 \tilde{J}_3 \\ \text{s.t.} \\ \text{C1: } \sum_{ST_y \in \text{Cache}(n,t)} D_{ST_y} \leq C_n, \quad \forall n \in n_{ALL}, \forall t \\ \text{C2: } \sum_{ST^* \in \Phi_n(t)} f_m(ST^*, t) \leq F_m(t), \quad \forall n \in n_{ALL}, \forall t \\ \text{C3: } T(k) \leq T_{\max}(k), \quad \forall k \in \{K_{\text{critical}}, K_{\text{normal}}\} \\ \text{C4: } E_{\text{total}}(n,t) \leq E_{\text{res}}(n,t), \\ \text{C5: } d_{ij} \leq d_{\max}(r_{ij}), \quad \forall (ij) \in E, \forall r_{ij} \in R \\ \text{C6: } DTR_{ij}(t) \geq D_{\text{req}}(ST^*), \quad \forall ST^* \in \text{Task}_{i \rightarrow j} \end{cases} \quad (34)$$

上述式为最小化问题的约束模型, 核心是为了通过调整权重, 平衡时延、缓存效率、能耗三个目标, 实现系统性能最优。 $\omega_1, \omega_2, \omega_3$ 为目标权重, 三者之和为1, 在权重设置方面, 本文采用网格搜索方法对权重参数进行标定。具体而言, 在满足权重和为1的约束下, 对不同权重组合进行遍历, 并通过仿真实验评估系统综合性能。最终选择能够在多指标之间取得较优平衡的权重组合作为实验设置。C1为缓存容量约束, 任意节点 m 在任意时刻 t , 其缓存的所有可缓存子任务的总数据量不能超过该节点的缓存容量上限。C2为计算资源约束, 任意节

点 m 在任意时刻 t , 分配给所有处理中子任务的计算资源总和不能超过该节点的剩余计算能力。C3为时延容忍约束, 任意任务的实际处理时延不能超过其最大容忍时延。C4为能源约束, 无人机、车载终端等能源受限节点, 在任意时刻 t 的总能耗不能超过其剩余能量。C5为通信链路距离约束, 任意节点之间的物理距离, 不能超过其通信链路类型的最大通信距离。C6为通信链路带宽约束, 节点 i 向节点 j 卸载子任务时, 链路的实际传输速率需满足该子任务的带宽需求。

4 GAT-MADRL 分布式缓存框架设计与实现

4.1 图拓扑演化算法

本文设计了基于距离阈值的拓扑重构规则, 当车载终端与RSU的距离超过通信半径, 图网络将删除原边并重新建立与新覆盖RSU的边。当UAV因任务调整飞行轨迹, 与LEO卫星和地面节点的链路质量低于阈值时, 触发边特征更新或拓扑重连。为兼顾实时性, 本文不在每个时隙对全图进行完全重建, 而采用事件触发的增量更新机制: 仅当节点位置变化超过距离阈值 R_{th} 、链路质量低于SNR阈值 Q_{th} 或缓存状态发生显著变化时, 才更新受影响节点及其一跳邻域。候选邻居搜索先按节点类型过滤, 再按链路质量保留前 K 个邻居, 从而减少无效遍历。因此, 图拓扑演化最坏情况下时间复杂度仍为 $O(n^2)$, 但常规单次更新复杂度可降为 $O(n_u K)$, 其中 n_u 为受影响节点数, K 为单节点保留的最大邻居数, 能够满足动态SAGVN场景下的在线构图需求。

4.2 多层注意力机制设计

在SAGVN架构中, 由于节点异构性与链路异构性, 传统的单一层级注意力无法同时适配节点特征和链路类型, 因此设计节点特征注意力与链路类型注意力的双层结构, 先筛选高价值节点, 再优选选择高可靠性链路。

4.2.1 节点特征注意力

SAGVN包含LEO、UAV、RSU、车载终端等异构节点, 若直接聚合所有邻居特征, 低资源节点的无效信息会干扰后续缓存决策。因此需通过节点特征注意力, 筛选出邻居节点, 提升特征质量。基于节点自身资源状态与邻居关联, 计算注意力权重, 让GAT优先关注高缓存容量、低能耗与高请

算法1 图拓扑演化算法

输入: 节点实时状态 $S(t)$ 、链路质量 $Q(t)$ 、 R_{th} 、 Q_{th}

输出: 动态有向异构图 $G(t+\Delta t)$

1. For 每个 SAGVN 节点 i :
2. 根据 $S(t)$ 更新节点特征向量
3. End For
3. For 每条 $G(t)$ 中的边:
4. 计算节点 i 到 j 的距离 d_{ij}
5. 计算链路信噪比 $SNR_{ij}=Q(t)[i,j]$;
6. If $d_{ij} > R_{th}$ or $SNR_{ij} < Q_{th}$;
7. 从 $G(t)$ 移除这条边
8. Else:
9. 根据 $Q(t)$ 更新边特征向量
10. End For
11. For 每个邻居不足的节点 i :
12. 找到候选节点 j , 满足 $d_{ij} \leq R_{th}$ and $SNR_{ij} \geq Q_{th}$
13. 将边 (i,j) 添加到 $G(t)$, 并初始化边特征向量
14. End For
15. Return $G(t+\Delta t)=G(t)$

求覆盖的节点, 具体实现流程如图2所示。

图2中, GAT在时隙 t 接收节点输入矩阵 $X(t)$ 和边特征矩阵 $Z(t)$ 。对于原始节点特征进行归一化处理之后, 再根据节点类型进行编码, 并使用共享线性层将低维特征映射到统一隐藏空间; 随后通过关联度计算与归一化确定每个邻居的特征占比, 按权重加权求和邻居高维特征, 得到最终注意力特征。其中关联度计算中拼接当前节点与邻居的高维特征, 用 LeakyReLU 激活函数计算关联度。

$$sco = \text{LeakyReLU} \left(\begin{matrix} M_a \times \text{concat}(feat_{cu}, feat_{nei}) \\ + b_a \end{matrix} \right) \quad (35)$$

其中, concat 表示特征拼接, M_a 为注意力系数矩阵, $feat_{cu}$ 为当前节点高维特征, $feat_{nei}$ 为有效邻居节点高维特征, b_a 为注意力偏置项。

4.2.2 链路类型注意力

对于通信链路, 用 r_{ij} 标记链路类型, q_{ij} 表示链路质量, 并进一步构造边特征 $z_{ij} = [r_{ij}, q_{ij}, b_{ij}, d_{ij}, \tau_{ij}, s_{ij}]$ 。其中, r_{ij} 通过权重矩阵映射为类型嵌入, b_{ij} 、 d_{ij} 、 τ_{ij} 和 s_{ij} 分别表示可用带宽、归一化距离、当前传输时延和链路稳定度; 所有连续边特征均进行归一化处理, 并根据 $Q(t)$ 实时更新。由此得到的链路特征 $feat2$ 再与节点特征注意力结果进行融合, 以量化异构链路差异。

在链路与节点特征的融合阶段, 把节点特征注意力 $feat1$ 和 $feat2$ 通过 concat 函数拼接, 通过 Lea-

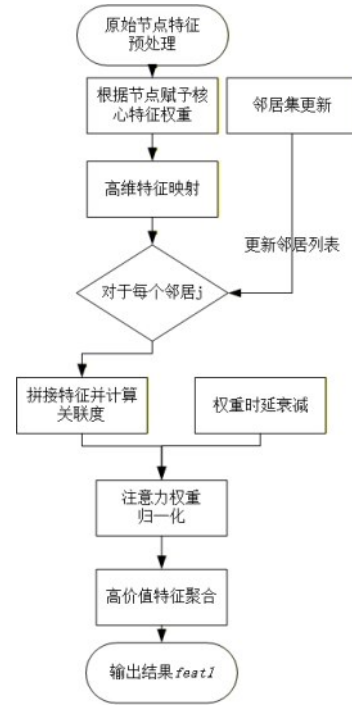


图2 特征流程图

kyReLU 函数计算链路关联度最后归一化得到链路权重 L_w , 最终融合特征 F_{fuse} 通过使用 L_w 加权邻居节点特征 F_{node_j} , 用 ReLU 激活得到。

$$F_{fuse} = \text{ReLU} \left(\sum L_w \times F_{node_j} \right) \quad (36)$$

由于单组注意力容易受干扰, 本文用多个独立注意力头 A_{head} 并行计算, 每个头输出 A_{out} 用平均池化融合所有。

$$A_{avg} = \frac{A_{out1} + A_{out2} + \dots + A_{outi}}{i} \quad (37)$$

4.3 GAT-MADRL 算法策略

本文将 GAT 作为智能体感知网络状态的编码器, MADRL 作为分布式决策器, 来弥补传统缓存策略的适配缺陷。

4.3.1 状态空间

每个智能体 m 在时隙 t 的状态 $State_m(t)$ 可以表示为

$$State_m(t+1) = \left[\begin{matrix} type_m, C_m^{rem}(t), F_m^{rem}(t), E_m^{rem}(t), \\ h_m^{GAT}(t), id_*, d_*, C_{cycle}, \rho_*, \tau_*^{max} \end{matrix} \right] \quad (38)$$

$State_m(t)$ 中的变量定义如下: $type_m$ 为类型编码; $C_m^{rem}(t)$ 为剩余缓存容量; $F_m^{rem}(t)$ 为剩余计算资源; $E_m^{rem}(t)$ 为剩余能量; $h_m^{GAT}(t)$ 为节点 m 通过 GAT 聚合邻居特征得到的特征向量; id_* 为内容标识; d_* 为子任务的数据量; C_{cycle} 为计算需求周期

数； ρ_* 为子任务的优先级系数； τ_*^{\max} 为子任务的最大容忍延迟。

4.3.2 动作空间

每个智能体在时隙 t 的动作 $a_m(t)$ 是一个离散决策变量，代表针对当前待处理子任务的联合缓存与卸载决策，动作空间设计如下。

$$a_m(t) = \begin{cases} 0, \text{本地执行, 不缓存} \\ 1, \text{本地执行, 缓存} \\ k, \text{子任务卸载至第 } k - 1 \text{ 个邻居节点} \end{cases} \quad (39)$$

4.3.3 奖励函数

智能体 m 时隙 t 执行动作 $a_m(t)$ 后，环境反馈的即时奖励 $r_m(t)$ 是一个多目标融合的标量值。

$$r_m(t) = \beta_1 \cdot r_{\text{delay}} + \beta_2 \cdot r_{\text{hit}} - \beta_3 \cdot \frac{E_*(t)}{E_m^{\text{rem}}(t)} \quad (40)$$

其中 $\beta_1, \beta_2, \beta_3$ 分别为权重系数，其取值同样通过网格搜索方法确定，并且三者之和为 1。 r_{hit} 为缓存命中率奖励，其值为 0 或 1。 r_{delay} 为时延奖励，其表达式如下。

$$r_{\text{delay}} = \begin{cases} -\log\left(\frac{T_*(t)}{\tau_*^{\max}}\right), & \text{if } T_*(t) \leq \tau_*^{\max} \\ -\zeta, & \text{otherwise} \end{cases} \quad (41)$$

其中 ζ 为超时惩罚常数。

4.3.4 GAT-MADRL 网络结构设计

为了实现分布式缓存策略，本文提出了 GAT-MADRL 框架，采用编码器、决策器、执行器三层结构，整体架构如图 3 所示。GAT 编码器用于从 SAGVN 有向异构图中提取节点特征，通过多头注

意力机制捕获节点间的依赖关系，从而生成具有拓扑感知能力的特征表示。在此基础上，构建基于 MADRL 的分布式决策模块，由多个智能体根据局部观测信息与全局聚合特征协同完成缓存决策。具体而言，GAT 编码器采用 4 个注意力头，每个注意力头的隐藏维度为 16，经拼接后形成 64 维节点特征表示。Actor 网络采用两层全连接结构，隐藏层维度分别为 128 和 64。在训练阶段，引入集中式 Critic 进行评估。Critic 以全局状态与联合动作的拼接向量为输入，采用三层全连接网络结构输出价值函数，并通过策略梯度方法对模型参数进行优化，以实现多智能体间的高效协同。

GAT-MADRL 更新策略如算法 2 所示。

4.3.5 复杂度分析

设 $|V_t|$ 和 $|E_t|$ 分别表示时隙 t 动态图中的节点数和边数， H 表示 GAT 注意力头数， d_h 表示单头隐藏维度， M 表示参与决策的智能体数量， B 表示经验回放训练的批量大小。则在单时间步内，图拓扑增量更新的平均时间复杂度为 $O(n_{uk})$ ，最坏情况下为 $O(n^2)$ ；GAT 编码器在稀疏图上的计算复杂度可表示为 $O(H|E_t|d_h + |V_t|Hd_h)$ ；所有智能体 Actor 前向决策的复杂度为 $O(MC_A)$ ，集中式 Critic 参数更新复杂度为 $O(BC_C)$ 。因此，GAT-MADRL 在单时间步的平均时间复杂度可近似表示为 $O(n_{uk} + H|E_t|d_h + MC_A + BC_C)$ ，最坏情况下为 $O(n^2 + H|E_t|d_h + MC_A + BC_C)$ 。若总训练轮数为 E 、每轮包含 T 个时间步，则整体训练时间复杂度为

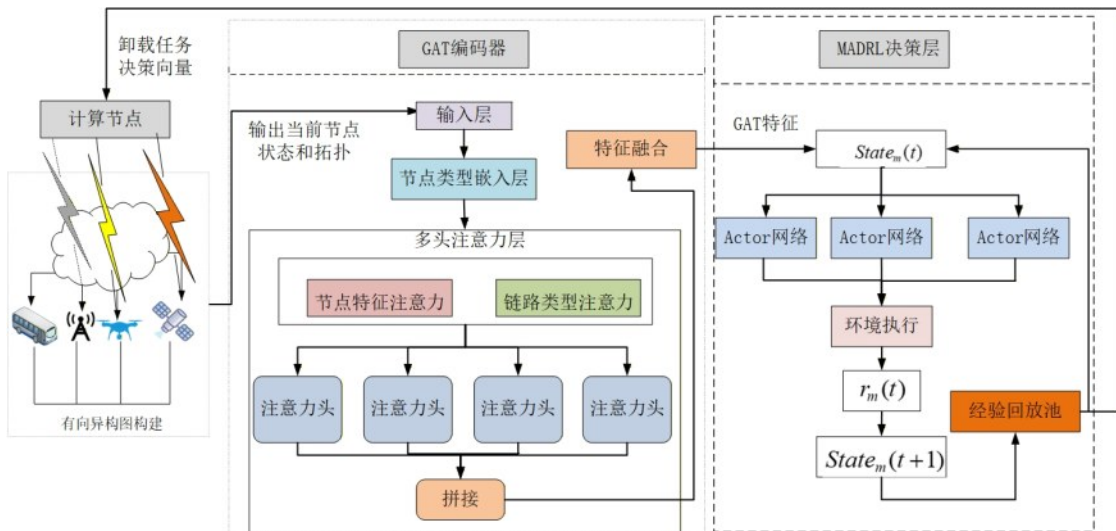


图 3 GAT-MADRL 架构

算法2 GAT-MADRL 更新策略

输入：经验缓冲池、GAT 编码器、Actor 网络、节点集合、折扣因子、软更新参数

输出：训练完成的 Actor 网络、GAT 编码器、每轮性能指标

1. 初始化各参数
2. **For** episode:
3. 初始化环境, 获取初始有向异构图和节点初始状态
4. **For** 每个时间步循环:
 5. GAT 编码: 处理图网络, 得到节点特征矩阵
 6. 根据节点类型提取局部状态
 7. 从特征向量中获取节点 i 的 GAT 特征
 8. 调用 Actor 网络生成动作概率分布, 执行动作, 获得即时奖励
 9. 将状态、动作、奖励放入经验回放池
 10. 执行所有智能体的联合动作, 更新环境状态, 生成下一时刻图
 11. 更新 Actor、GAT 编码器, 目标网络软更新
 12. 调用算法 1 图拓扑演化算法更新图
13. **End For**
14. 计算本轮平均奖励、缓存命中率、时延、能耗等指标
15. **If** episode%100==0:
16. 保存模型参数, 输出性能指标
17. **End If**
18. **End For**
19. **Return** 训练完成的 Actor、GAT 编码器、每轮指标

$$O\left(ET\left(n_{uk} + H|E_i|d_h + MC_A + BC_C\right)\right).$$

空间开销主要由动态图存储、模型参数和经验回放池三部分构成。其中, 节点与边特征存储复杂度分别为 $O(|V_i|d_x)$ 和 $O(|E_i|d_z)$, 本文中 $d_x = 10$ 、 $d_z = 6$; GAT 编码器、各智能体 Actor 及集中式 Critic 的参数存储复杂度分别记为 $O(P_G)$ 、 $O(MP_A)$ 和 $O(P_C)$; 当经验回放池容量为 N_{buf} 时, 其空间复杂度为 $O(N_{buf}(d_s + d_a + 1))$ 。因此, 算法总体空间复杂度可表示为 $O(|V_i|d_x + |E_i|d_z + P_G + MP_A + P_C + N_{buf}(d_s + d_a + 1))$ 。

收敛复杂度方面, GAT-MADRL 属于含图编码器的多智能体非凸优化模型, 难以给出严格的闭式全局收敛上界, 因此本文从训练迭代量级和方法开销差异两个角度进行分析。在固定批量大小 B 和软更新系数 τ 的条件下, 每轮训练的收敛代价与时间步数 T 及每步参数更新开销近似线性相关, 总体收敛复杂度可近似表示为 $O(ETB(H|E_i|d_h + MC_A + C_C))$ 。与 Random 策略相比, 本文方法在线决策开销更高, 但能够实现面向缓存命中率、时延和能耗

的联合优化; 与 MADQN (Multi-Agent Deep Q-Network)、MADDPG (Multi-Agent Deep Deterministic Policy Gradient) 等不建模图结构的方法相比, 本文额外引入了 GAT 编码开销 $O(H|E_i|d_h)$, 但减少了各智能体对冗余邻居信息的重复建模, 能够在动态拓扑下获得更稳定的策略表示; 与 GA (Genetic Algorithm) 这类集中式搜索方法相比, 其无需在每次决策中执行多代种群迭代, 避免了随节点规模增长而快速扩大的组合搜索开销, 因此更适用于 SAGVN 场景中的在线缓存决策。

5 仿真实验及分析

5.1 实验设置

为验证本文框架在 SAGVN 缓存决策中的有效性, 构建了仿真实验平台, 仿真过程中考虑网络的动态特性, 车载终端按照随机速度模型移动, UAV 根据任务需求动态调整飞行位置, LEO 卫星按照轨道周期更新位置, 从而形成时变拓扑结构。配置如表 3、表 4 所示。

表 3 实验环境配置

| 名称 | 信息 |
|--------|---------------------------|
| 操作系统 | Windows 11 |
| CPU | Intel i9-12900HX |
| GPU | NVIDIA GeForce RTX 4060 |
| 开发语言 | Python 3.9 |
| 深度学习框架 | PyTorch 2.3.1 + CUDA 12.1 |

5.2 仿真结果

为验证权重设置的合理性, 本文进一步分析不同权重组合对系统性能的影响, 如图 4、图 5 所示。

图 4 中的即时奖励值为实验后续趋于收敛的值, 图 5 中的综合得分为 $-J$ 。实验结果表明, 当 $\beta_1 = 0.5$, $\beta_2 = 0.3$, $\beta_3 = 0.2$ 时, 系统在即时奖励值取得最优, 当 $\omega_1 = 0.5$, $\omega_2 = 0.3$, $\omega_3 = 0.2$ 时, 系统在时延、能耗与缓存命中率之间取得较优平衡, 因此选取这两组权重作为最终参数。

学习率会影响算法收敛效率与优化性能。过大易引发训练震荡, 难稳收至全局最优甚至发散; 过小则训练停滞。为此, 本文设计不同学习率的对比实验, 以筛选适配 SAGVN 缓存优化的最优学习率, 如图 6 所示。

结合实际需求, 最终选用 $1e-2$ 为后续实验的学

表4 仿真参数配置

| 参数 | 值 |
|--------------|---------------|
| UAV 高度 | 0.5 km |
| 最大容忍时延 | 0~2 s |
| 任务数据量 | 0~500 MB |
| 内容有效时长 | 10s~1000 s |
| 车载终端行驶速度 | 18~72 km/h |
| 车载终端缓存容量 | 50~100 MB |
| RSU 缓存容量 | 500~1000 MB |
| UAV 缓存容量 | 300~500 MB |
| LEO 卫星缓存容量 | 1000~2000 MB |
| 车载终端发射功率 | 3~5 W |
| RSU 发射功率 | 80 W |
| UAV 发射功率 | 90 W |
| UAV 最大通信距离 | 1~5 km |
| RSU 最大通信距离 | 0.5~1 km |
| GAT 注意力头数 | 4 |
| GAT 单头隐藏维度 | 16 |
| GAT 输出特征维度 | 64 |
| Actor 网络隐藏层 | 2层(128, 64) |
| Critic 网络结构 | 3层(256-128-1) |
| 批量大小 | 64 |
| 经验回放池大小 | 100000 |
| 软更新系数 τ | 0.01 |
| 折扣因子 | 0.95 |

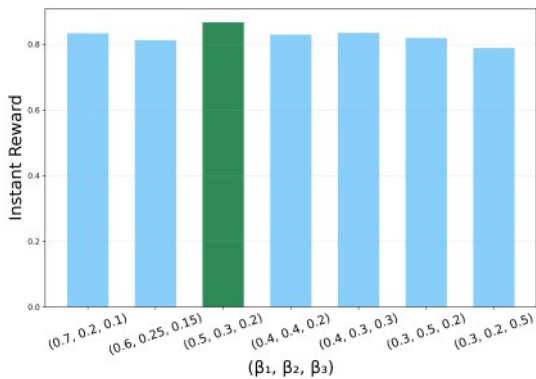


图4 权重对奖励值的影响

习率。该学习率在训练初期能够使奖励快速上升，有助于算法迅速适应动态变化的环境。

为了验证本文算法方案的有效性，在实验中与以下四种策略进行了对比。

(1) 随机策略：车辆的卸载决策、缓存的策略以及资源分配均为随机。

(2) MADDPG 策略：每个需要做决策的节点被建模为一个智能体，每个智能体遵循 DDPG 算法^[27-29]。

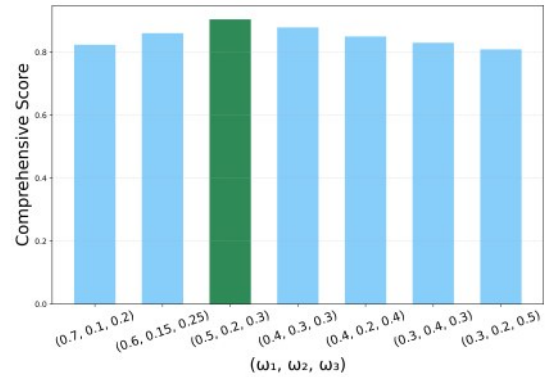


图5 权重对综合得分的影响

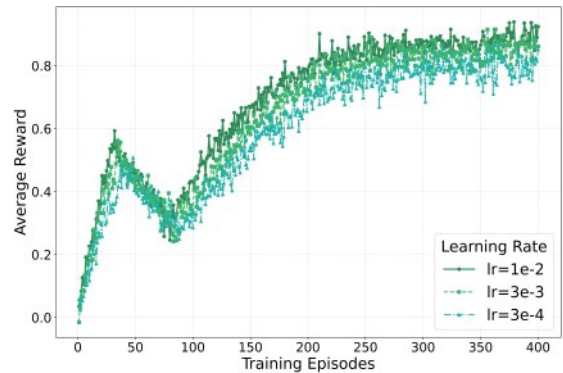


图6 学习率变化对奖励值的影响

(3) MADQN 策略：每个需要做决策的节点被建模为一个独立的智能体，每个智能体遵循基于值函数的 DQN 算法^[30-31]。

(4) GA 策略：该策略将整个网络的缓存与卸载联合优化问题建模为一个集中式的组合优化问题。

所提算法方案的奖励值训练曲线如图7所示。

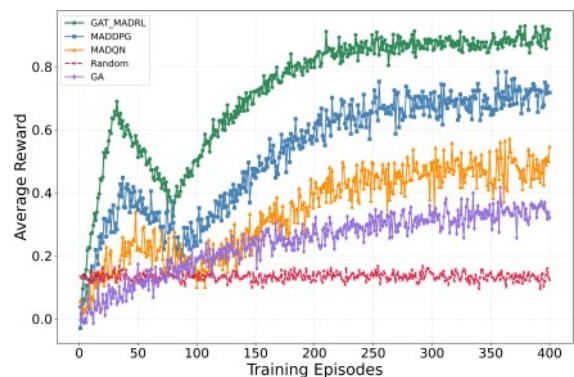


图7 训练代数变化对奖励值的影响

从图7可以看出，本文所提算法的性能表现良好。具体而言，该算法的平均奖励值上升速率更快，在训练后期仍可稳定维持在较高的奖励水平；

同时，所提算法可在有限的训练轮次内完成收敛，且收敛后波动幅度极小，表明该方法在动态环境中具备优异的稳定性能。

任务数量变化对时延的影响如图8所示。

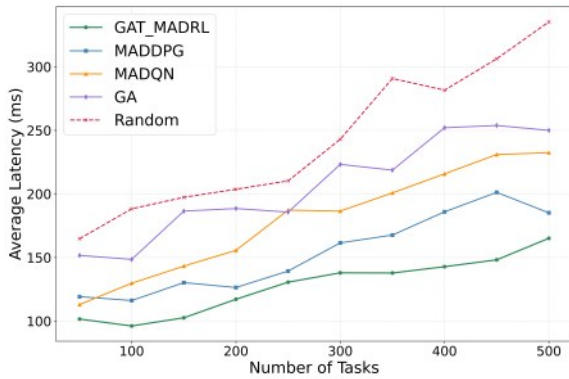


图8 任务数量变化对时延的影响

图8显示，随着任务数量从100增加到500，各算法处理时延均上升，而本文算法策略时延始终较低，相比MADDPG、MADQN、GA、Random算法策略分别降低22.12%、26.83%、38.12%、51.61%。此外，在SAGVN动态环境中，车载终端的高速移动、UAV飞行轨迹调整以及LEO卫星的轨道运动会导致节点间链路状态频繁变化，使得任务请求节点与缓存节点之间的最优匹配关系不断重构。在任务数量增加的情况下，这种动态拓扑变化进一步加剧了链路拥塞与传输不稳定，从而导致时延上升更加明显。相比之下，本文方法通过GAT对动态拓扑进行实时建模，能够自适应捕捉链路变化并调整缓存与卸载策略，因此在高动态场景下仍能保持较低时延。

任务数量变化对能耗的影响如图9所示。

图9显示，随着任务数量从100增加到500，各算法平均能耗均上升，而本文算法策略能耗始终较

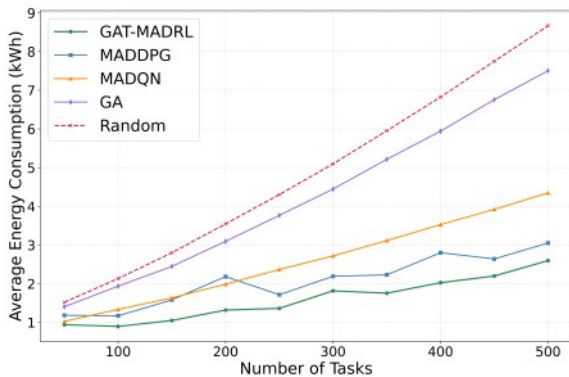


图9 任务数量变化对能耗的影响

低，相比MADDPG、MADQN、GA、Random算法策略分别降低24.21%、43.23%、67.53%、71.68%。同时，在动态SAGVN环境中，节点移动与链路波动会导致数据传输路径频繁变化，增加重复传输与无效计算，从而引起额外能耗开销。尤其是在任务数量增加时，动态拓扑带来的路径不稳定性会进一步放大能耗增长趋势。而本文方法能够通过图注意力机制优先选择稳定且高质量链路，并动态优化任务分配，从而有效降低因拓扑变化带来的额外能耗。

任务数量变化对缓存命中率的影响如图10所示。

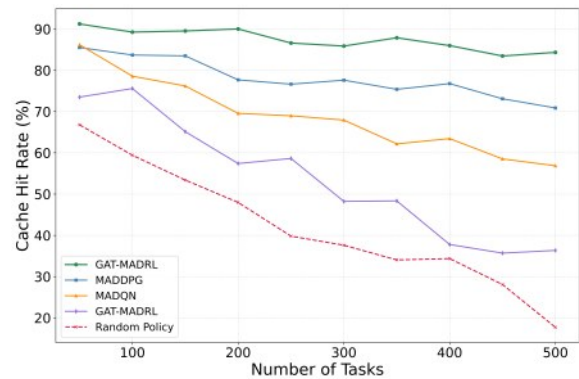


图10 任务数量变化对缓存命中率的影响

图10显示，随着任务数量从100增加到500，各算法缓存命中率均下降，本文算法策略命中率始终较高，相比MADDPG、MADQN、GA、Random算法策略分别提升6.21%、21.43%、49.11%、261.19%，能更高效提升缓存资源利用率。在SAGVN动态环境下，车载终端的移动以及UAV与卫星位置变化会改变内容需求的空间分布，使得原有缓存内容与当前请求之间的匹配关系不断变化，从而降低缓存命中率。随着任务数量增加，这种“供需错位”现象更加明显。而本文方法通过GAT捕捉节点间的动态关联关系，并结合MADRL进行自适应缓存决策，使缓存内容能够更好地匹配动态变化的需求分布，从而在高动态环境中仍保持较高的缓存命中率。

任务数量变化对综合得分的影响如图11所示。

图11显示，随着任务数量从100增加到500，各算法综合得分均下降，本文算法策略综合得分始终较高，相比MADDPG、MADQN、GA、Random

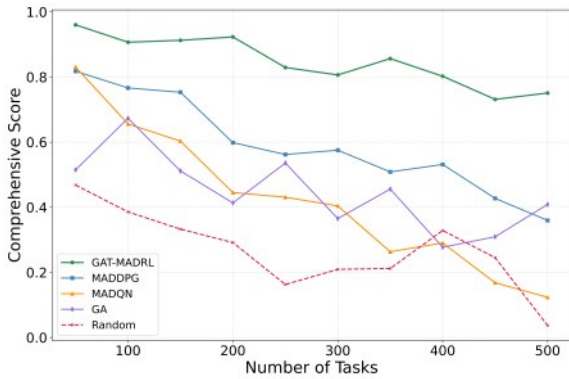


图11 任务数量变化对综合得分的影响

算法策略分别提升 50.00%、87.52%、150.00%、400.00%，能更全面优化任务处理综合性能，在动态环境中实现多目标协同优化，表现出更强的鲁棒性和适应能力。

内容有效时长作为缓存内容的核心属性之一，直接决定缓存资源的复用价值与生命周期管理效率。如图12所示。

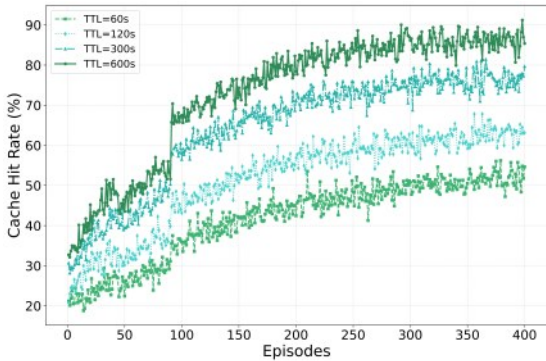


图12 内容有效时长对缓存命中率的影响

从图12可看出：延长内容有效期能够提升缓存命中率，这是因为若过早删除了受欢迎但已过期的内容，会带来明显的性能损耗，增加缓存未命中。

为验证本文框架中多层注意力机制的有效性，本文设计了针对性的融合对比实验如下：

(1) 仅节点特征注意力 (Node-Only)：仅保留框架中的节点特征注意力机制，通过节点资源状态筛选高价值邻居，不考虑链路质量差异；

(2) 仅链路类型注意力 (Link-Only)：仅保留链路类型注意力机制，优先适配高质量通信链路，未对异构节点的资源约束进行差异化筛选；

(3) 无注意力 (No-Attention)：移除所有注意力机制，采用均等权重聚合邻居节点与链路特征，

模拟传统图网络的特征处理方式。

任务数量变化对不同注意力配置下平均时延变化对比图、能耗变化对比图、缓存命中率变化对比图和综合得分变化对比图如图13、图14、图15、图16所示。

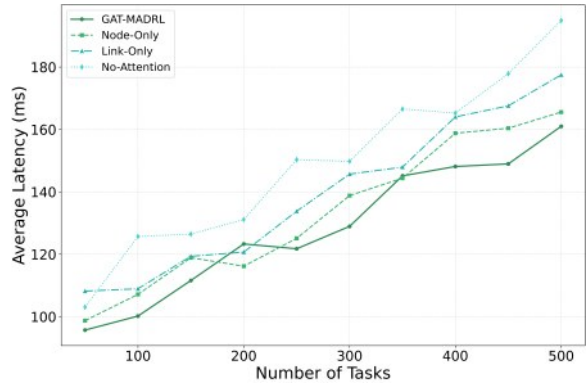


图13 不同注意力配置下时延影响

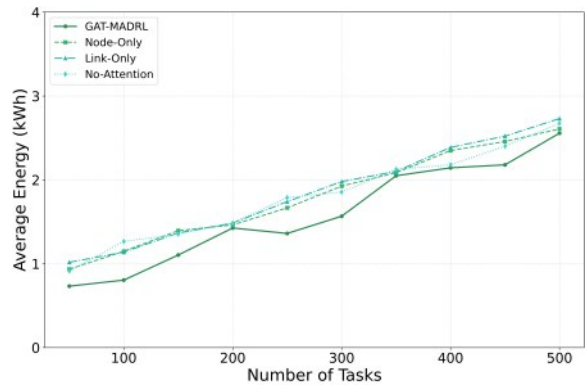


图14 不同注意力配置下能耗影响

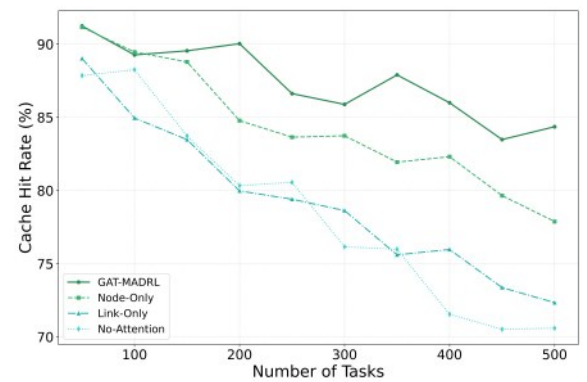


图15 不同注意力配置下缓存命中率影响

综合图13、图14、图15和图16可知：随着任务数量从100增加到500，不同注意力配置下的各项性能呈现不同变化趋势，对于时延和能耗总体呈上升趋势，缓存命中率总体呈下降趋势，综合得分总体呈波动变化趋势；而本文所提框架在各指标上

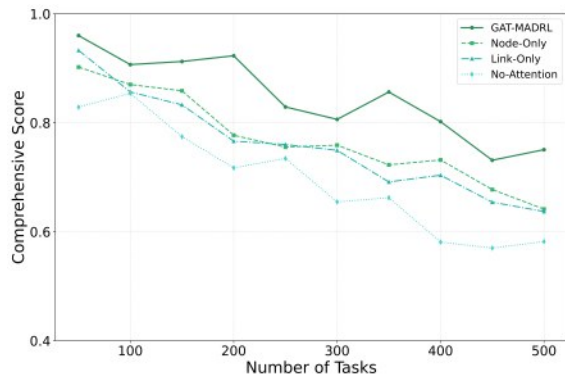


图16 不同注意力配置下综合得分的影响

均表现更优，其指标始终维持在较低或合理水平，缓存命中率始终保持较高水平，相比仅节点注意力、仅链路注意力及无注意力的配置优势显著。

6 结束语

SAGVN作为未来智能交通系统的关键支撑，面临节点异构、拓扑动态及缓存协同复杂等挑战。本文提出GAT与MADRL融合的GAT-MADRL策略，构建SAGVN有向异构图，设计融合框架利用GAT感知节点与链路重要性、MADRL实现分布式协同优化，降低通信开销的同时实现多目标均衡。仿真表明，该策略相较于对比方法，性能更优，验证了其在动态异构环境下的有效性、优越性与鲁棒性。

本研究未完全复现实际场景的严苛条件。未来工作将聚焦于纳入更复杂的业务异构性与严苛网络约束，以增强其工程适用性与推广价值。

参考文献：

- [1] Rajkumar Y, Santhosh Kumar S V N. A comprehensive survey on communication techniques for the realization of intelligent transportation systems in IoT based smart cities[J]. Peer-to-Peer Networking and Applications, 2024, 17(3): 1263-1308.
- [2] Ju Y, Gao Z, Wang H, et al. Energy-efficient cooperative secure communications in mmWave vehicular networks using deep recurrent reinforcement learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(10): 14460-14475.
- [3] Moghaddasi K, Rajabi S, Gharehchopogh F S. An enhanced asynchronous advantage actor-critic-based algorithm for performance optimization in mobile edge computing-enabled internet of vehicles networks[J]. Peer-to-Peer Networking and Applications, 2024, 17(3): 1169-1189.
- [4] 张鸿, 廖彧歆, 王汝言, 等. 面向密集场景的空天地网络资源分配算法[J]. 电子与信息学报, 2024, 46(5): 1968-1976.
- [5] Zhang Hong, Liao Yuxin, Wang Ruyan, et al. Resource Allocation Algorithm for Space-Air-Ground Networks in Dense Scenarios[J]. Journal of Electronics and Information Technology, 2024, 46(5): 1968-1976.
- [5] 牛志升, 张钦宇, 唐余亮. 面向沉浸式体验的空天地一体化车联网体系架构与关键技术[J]. 物联网学报, 2024, 1(2): 17-27.
- Niu Zhisheng, Zhang Qinyu, Tang Yuliang. Architecture and Key Technologies of Space-Air-Ground Integrated Vehicle Networking for Immersive Experience[J]. Chinese Journal of Internet of Things, 2024, 1(2): 17-27.
- [6] Liang H, Yang Z, Zhang G, et al. Resource allocation for space-air-ground integrated networks: A comprehensive review[J]. Journal of Communications and Information Networks, 2024, 9(1): 1-23.
- [7] Li S, Liu G, Li L, et al. A review on air-ground coordination in mobile edge computing: Key technologies, applications and future directions[J]. Tsinghua Science and Technology, 2024, 30(3): 1359-1386.
- [8] 赵雄文, 张钰, 秦鹏, 等. 空天地一体化无线光通信网络关键技术及其发展趋势[J]. 电子学报, 2022, 50(1): 1-17.
- Zhao Xiongwen, Zhang Yu, Qin Peng, et al. Key Technologies and Development Trends of Space-Air-Ground Integrated Wireless Optical Communication Networks[J]. Acta Electronica Sinica, 2022, 50(1): 1-17.
- [9] 张宗辉, 秦祥宇, 朱思峰, 等. 基于MADRL的海上物联网任务卸载优化方案[J]. 物联网学报, 1-19.
- Zhang Zonghui, Qin Xiangyu, Zhu Sifeng, et al. MADRL-Based Task Offloading Optimization Scheme for Maritime Internet of Things[J]. Chinese Journal on Internet of Things, 1-19.
- [10] Gang Y, Zhang Y, Zhuo Z. Joint task offloading and resource allocation strategy for space-air-ground integrated vehicular networks [J]. Tsinghua Science and Technology, 2024, 30(3): 1027-1043.
- [11] 徐煜凯, 徐波, 张凌豪, 等. 基于图学习的空天地一体化网络合作推理方法研究[J]. 物联网学报, 1-17.
- Xu Yukai, Xu Bo, Zhang Linghao, et al. Research on Cooperative Inference Method of Space-Air-Ground Integrated Network Based on Graph Learning[J]. Chinese Journal on Internet of Things, 1-17.
- [12] Qi X, Chong J, Zhang Q, et al. Toward cooperatively caching in multi-UAV-assisted network: A queue-aware CDS-based reinforcement learning mechanism with energy-efficiency maximization[J]. IEEE Internet of Things Journal, 2024, 11(9): 16461-16477.
- [13] Wu C, Xu Z, He X, et al. Proactive caching with distributed deep reinforcement learning in 6g cloud-edge collaboration computing [J]. IEEE Transactions on Parallel and Distributed Systems, 2024, 35(8): 1387-1399.
- [14] LI Y L, CHAI Z Y, TAN F, et al. Temporal Data Scheduling in Internet of Vehicles Using an Improved Decomposition Based Multi-Objective Evolutionary Algorithm [J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(5): 5282-5295.
- [15] 周旭, 李泰新, 覃毅芳, 等. 基于边缘智能协同的天地一体化信息网络研究[J]. 电信科学, 2024, 36(7): 71-79.

- Zhou Xu, Li Taixin, Qin Yifang, et al. Research on Space-Ground Integrated Information Network Based on Edge Intelligence Collaboration[J]. Telecommunications Science, 2024, 36(7): 71-79.
- [16] Ji S, Zhou D, Sheng M, et al. Dynamic space-ground integrated mobility management strategy for mega LEO satellite constellations[J]. IEEE Transactions on Wireless Communications, 2024, 23(9): 11043-11060.
- [17] Dai M, Huang N, Wu Y, et al. Unmanned-aerial-vehicle-assisted wireless networks: Advancements, challenges, and solutions[J]. IEEE Internet of Things Journal, 2022, 10(5): 4117-4147.
- [18] 吴红海, 王白冰, 马华红, 等. 移动车载边缘网络中基于递归深度强化学习的协作缓存接力算法[J]. 通信学报, 2024, 45(11): 277-286.
- Wu Honghai, Wang Baibing, Ma Huahong, et al. Cooperative Caching Relay Algorithm Based on Recursive Deep Reinforcement Learning in Mobile Vehicular Edge Networks[J]. Journal on Communications, 2024, 45(11): 277-286.
- [19] Jin Z, Song T, Jia W K. An adaptive cooperative caching strategy for vehicular networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(10): 9502-9517.
- [20] Chen G, Duan W, Sun J, et al. Social-aware assisted edge collaborative caching based on deep reinforcement learning joint with digital twin network in Internet of Vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(10): 14785-14802.
- [21] Yoo S, Jeong S, Kim J, et al. Cache-assisted mobile-edge computing over space - air - ground integrated networks for extended reality applications[J]. IEEE Internet of Things Journal, 2024, 11(10): 18306-18319.
- [22] Yin F, Liu Q, Liu D, et al. Joint coded caching and resource allocation for multimedia service in space-air-ground integrated networks[J]. IEEE Transactions on Communications, 2024, 72(11): 6839-6853.
- [23] Zhang P, Li Y, Kumar N, et al. Distributed deep reinforcement learning assisted resource allocation algorithm for space-air-ground integrated networks[J]. IEEE Transactions on Network and Service Management, 2022, 20(3): 3348-3358.
- [24] Y. Ran, Y. Ding, S. Chen, et al. Fully-Distributed Dynamic Packet Routing for LEO Satellite Networks: A GNN-Enhanced Multi-Agent Reinforcement Learning Approach[J], IEEE Transactions on Vehicular Technology, 2025, 74(3): 5229-5234.
- [25] Wang P, Zhang J, Zhang X, et al. Convergence of satellite and terrestrial networks: A comprehensive survey[J]. IEEE Access, 2019, 8: 5550-5588.
- [26] D. Li, H. Zhang, H. Ding, et al., User-Preference-Learning-Based Proactive Edge Caching for D2D-Assisted Wireless Networks[J], IEEE Internet of Things Journal, 2023, 10(13): 11922-11937.
- [27] He X, Lu H, Du M, et al. QoE-based task offloading with deep reinforcement learning in edge-enabled Internet of Vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(4): 2252-2261.
- [28] Wei R, Qin T, Huang J, et al. Resource allocation scheduling scheme for task migration and offloading in 6G Cyber twin internet of vehicles based on DRL[J]. IET Communications, 2024, 18(18): 1244-1265.
- [29] Aishwarya R, Vetriselvi V, Meignanamoorthi D. A Review on Computational Optimization Strategies and Collaborative Techniques of Vehicular Task Offloading in the Era of Internet of Vehicles and 6G[J]. Edge of Intelligence: Exploring the Frontiers of AI at the Edge, 2025: 1-49.
- [30] Ji Y, Wang Y, Zhao H, et al. Multi-agent reinforcement learning resources allocation method using dueling double deep Q-network in vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2023, 72(10): 13447-13460.
- [31] Moghaddasi K, Rajabi S, Gharehchopogh F S. Multi-objective secure task offloading strategy for block-chain-enabled IoV-MEC systems: a double deep Q-network approach[J]. IEEE Access, 2024, 12: 3437-3463.