

# 联邦深度强化学习的车联网任务卸载优化

李振文<sup>1</sup> 武鹏荣<sup>1</sup> 雷 强<sup>1</sup>

LI Zhenwen WU Pengrong LEI Qiang

## 摘 要

随着车联网和移动边缘计算（MEC）的快速发展，动态任务卸载与资源分配成为提升系统性能和用户体验的关键挑战。传统方法依赖静态模型或集中式训练，难以适应高动态车联网环境，且面临数据隐私与资源分配不均衡等问题，为此，文章设计了一种基于联邦学习与深度强化学习的协同优化框架（FL-MADDPG），结合多智能体深度确定性策略梯度算法与联邦学习机制，以解决车联网中的任务卸载与资源分配问题，通过联邦学习保护用户数据隐私，同时利用多智能体强化学习实现分布式协作优化，动态调整卸载决策与资源分配策略。仿真实验表明，与传统算法（如 DDPG、DQN）相比，FL-MADDPG 在资源受限场景下任务完成率提升 15% 以上，延迟降低 50%，能耗减少 40%，并展现出对动态任务分布与设备规模扩展的强鲁棒性，为车联网的高效资源管理与隐私保护提供了理论支持与技术路径。

## 关键词

移动边缘计算；任务卸载；联邦学习；深度强化学习；动态环境适应性

doi: 10.3969/j.issn.1672-9528.2025.10.004

## 0 引言

随着车联网与移动边缘计算的深度融合，动态环境下的任务卸载与资源分配面临着越来越复杂的挑战。传统静态模型与启发式方法难以适应车辆高速移动、网络状态不稳定等场景需求，急需更具自适应性、实时性和安全性的解决方案。近年来，深度强化学习技术的发展为动态决策提供了新方向：基于深度神经网络的卸载决策框架通过优化二元选择，显著降低了计算的复杂度<sup>[1]</sup>；数字孪生网络与异步强化学习的结合，有效减少了系统的长期成本<sup>[2]</sup>；多智能体协作方法通过建模任务依赖关系，提升了任务调度的效率<sup>[3]</sup>；而分布式算法在异构资源分配中展现出优化潜力<sup>[4-5]</sup>。此外，安全卸载与资源联合优化的研究进一步增强了系统的可靠性与隐私保护能力<sup>[6-7]</sup>。这些成果为车联网任务卸载提供了多样化的技术方向，但仍然需要突破多重核心瓶颈。

目前研究的主要局限体现在四个方面：

一是隐私保护机制的鲁棒性不足，尽管联邦学习被引入以保障数据隐私，但其静态聚合策略难以适应车辆频繁加入或退出的动态场景，数据异质性引发的模型偏移问题加剧了隐私泄露风险；

二是动态环境响应滞后，集中式训练的深度强化学习模型更新延迟显著，数字孪生环境与实际场景的同步存在偏差，导致策略决策无法实时、准确地匹配快速变化的环境状态；

三是多智能体协作效率低下，随着智能体数量增加，通信开销呈指数级增长，现有框架缺乏对通信带宽约束的考量，策略协调复杂度限制了大规模车联网的扩展性；

四是资源分配失衡问题突出，高峰期边缘服务器负载过载与车辆本地资源闲置的矛盾显著，传统算法未能充分挖掘跨层资源协同的潜力，系统整体效能受限。

针对上述的挑战，本研究设计的解决方案如下：

（1）构建联邦学习与深度强化学习的双驱动框架，通过动态聚合策略增强模型对网络拓扑变化的适应性，实现隐私数据的本地化高效处理；

（2）设计分布式多智能体优化算法，结合时空注意力机制，提升实时决策能力，显著缩短策略响应时间；

（3）开发智能中继优化机制，基于动态带宽感知与价值评估模型，减少冗余通信，提升协作效率与冲突检测精度；

（4）建立跨层资源调度模型，统筹车辆、边缘与云端资源，优化闲置资源利用率，降低任务过载风险。

实验验证表明，所提方案在系统效能、响应速度与资源均衡性等方面均优于传统方法，为智能交通系统的实时性保障、隐私安全强化与资源优化管理提供了理论支撑与技术范式。

本文的主要贡献为以下三点：

（1）将联邦学习与深度强化学习框架，设计出一种兼顾数据隐私与动态优化的新型框架，通过联邦学习保障本地数据安全，结合深度强化学习实现高效任务卸载与资源调度，

1. 西安工程大学 陕西西安 710600

有效缓解车联网场景中的延迟与能耗问题。

(2) 多智能体协作优化算法：基于多智能体深度强化学习，设计分布式算法框架，通过联邦协同训练增强模型鲁棒性与泛化能力，解决多智能体竞争协作与动态环境适配的复杂性挑战。

(3) 系统性实验验证与性能提升：通过仿真实验验证算法在降低任务延迟、提升完成率及优化能耗方面的显著优势，为车联网技术提供兼具理论价值与实践指导的创新方法。

## 1 相关工作

近年来，联邦学习（federated learning, FL）作为分布式机器学习范式，为移动边缘计算（mobile edge computing, MEC）系统中的数据隐私保护与协同训练提供了新思路。针对分布式训练中数据非独立同分布（Non-IID）与设备异构性挑战，McMahan 等人<sup>[8]</sup>提出的 FedAvg 算法通过本地参数加权聚合，在降低通信开销的同时保障模型收敛性；Wang 等人<sup>[9]</sup>进一步从理论层面分析 FL 在 MEC 中的收敛边界，并设计资源受限条件下的本地-全局更新平衡策略。在数据安全方面，Lu 等人<sup>[10]</sup>结合混合区块链架构与 FL 技术，通过两阶段验证机制确保数据可靠性；Li 等人<sup>[11]</sup>则融合深度  $Q$  网络与 FL，构建车联网环境下的高效安全数据共享方案。此外，Samarakoon 等人<sup>[12]</sup>针对设备异构性设计异步联邦学习框架，有效提升模型训练效率。这些研究表明，FL 与 MEC 的深度融合可协同优化通信负载、模型性能与隐私保护能力。

虽然现有成果在资源分配、任务卸载及联邦学习等方面取得显著进展，但仍存在以下局限：

(1) 异构任务在端-边-云分层架构中的差异化卸载需求尚未被充分建模，难以实现多目标协同优化；

(2) 传统强化学习算法在动态车联网环境中面临策略泛化性不足的问题，且集中式训练模式易引发数据隐私风险。

对此，本文设计了基于联邦学习的多智能体协同优化框架，通过构建端-边-云分层任务卸载模型，结合 MADDPG 算法与异步联邦训练机制，实现异构资源分配与隐私保护的双重优化，为车联网服务提供新的技术路径。

## 2 系统模型

考虑一个多时隙多用户的动态车桩边缘计算系统，其核心目标是在 V2I 通信模型下实现任务卸载与资源分配的联合优化。

如图 1 所示，系统以时隙  $\Delta t$  为基本时间单元，场景设定为城市双向车道环境，车辆沿道路以速度  $V_{\text{car}}$ 。每个时隙内，车辆位置根据运动模型更新为  $l_j^{t+1}$ ，其更新公式为：

$$l_j^{t+1} = l_j^t + v_{\text{car}} \times \Delta t \quad (1)$$

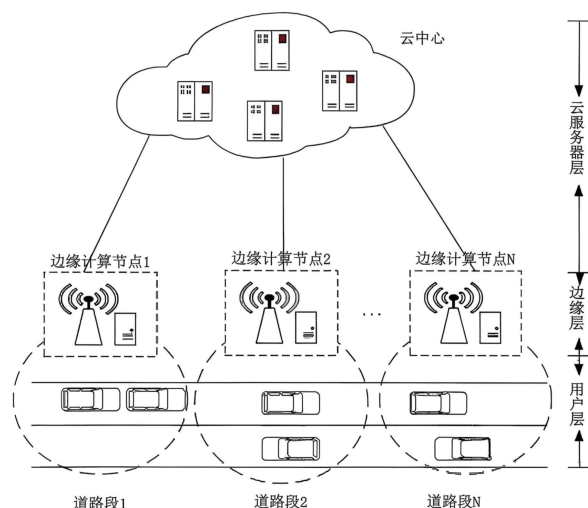


图 1 系统模型

同时车桩与边缘服务器（VEC）的覆盖关系随车辆移动动态调整。系统包含  $r$  个边缘服务器构成的集合  $S = \{S_1, S_2, \dots, S_p, \dots, S_r\}$ ，其中，每个服务器  $S_i$  由位置坐标  $(l_i, l_p)$ 、覆盖半径  $R$  及计算资源  $F$ （划分为  $\gamma$  类）共同定义。车辆集合  $V = \{V_1, V_2, \dots, V_h\}$ ，其中，每辆车配备车载通信单元，其本地计算资源同样受限于  $\gamma$  类划分。车辆仅能与覆盖范围内（距离满足  $\text{dis}(V_i, S_i) \leq R$ ）的边缘服务器交互，通过正交频分多址（OFDMA）技术分配独立信道进行任务传输，以避免多径干扰。系统通过动态优化车辆任务的本地执行、边缘卸载及资源分配策略，在满足低时延与低能耗约束的前提下，最大化任务完成率与资源利用率。

### 2.1 任务模型

任务到达过程服从泊松分布，到达速率为  $\lambda$ ，任务间隔时间满足指数分布  $T_{\text{arrival}} \sim \text{Exp}(\lambda)$ 。系统采用 M/M/m 优先级队列模型，高优先级任务优先处理以保障实时性需求。排队时延由队列中平均任务数  $\bar{N}_q$  与服务器资源动态决定，计算为  $T_{\text{queue}} = \bar{N}_q / (\mu \cdot m - \lambda)$ ，其中  $m$  为服务器数量； $\mu$  为单服务器处理速率。服务时延则与任务计算需求  $D$  及分配的资源相关，表达式为  $T_{\text{service}} = D / \mu$ ；若任务卸载至多服务器并行处理，服务时延可随  $m$  动态调整以优化效率。总时延为排队时延与服务时延的线性叠加，即  $T_{\text{total}} = T_{\text{queue}} + T_{\text{service}}$ ，此模型通过优先级调度机制平衡系统负载，在保障关键任务低延迟的同时，利用队列理论实现资源动态分配与整体效率优化。

### 2.2 计算模型

本小节构建了统一的任务卸载时延与能耗计算模型，通过参数化方法刻画不同卸载场景（本地执行、MEC 服务器、云中心及邻近节点）的动态特性。定义任务总时延为：

$$T_{\text{total}} = T_{\text{queue}} + \alpha \cdot T_{\text{trans}} + \beta \cdot T_{\text{comp}} \quad (2)$$

式中： $T_{\text{queue}}$  为排队时延，由服务器资源负载动态决定； $T_{\text{trans}}$

和  $T_{\text{comp}}$  分别为传输时延与计算时延;  $\alpha \in \{0,1\}$  和  $\beta \in \{0,1\}$  为场景选择系数。任务总能耗可建模为:

$$E_{\text{total}} = \gamma \cdot E_{\text{trans}} + \delta \cdot E_{\text{comp}} \quad (3)$$

式中:  $E_{\text{trans}}$  为传输能耗;  $E_{\text{comp}}$  为计算能耗;  $\gamma$  和  $\delta$  为能耗权重因子, 与卸载目标类型相关。具体而言:

(1) 本地执行:  $\alpha=0$  (无传输)、 $\beta=1$ 、 $\gamma=1$ 、 $\delta=1$ , 仅含本地计算时延  $T_{\text{comp}}$  与计算能耗  $E_{\text{comp}}$  的公式为:

$$T_{\text{comp}} = D_n / f_{\text{local}} \quad (4)$$

$$E_{\text{comp}} = k \cdot f_{\text{local}}^3 \cdot T_{\text{comp}} \quad (5)$$

(2) MEC 卸载:  $\alpha=1$ 、 $\beta=1$ 、 $\gamma=1$ 、 $\delta=1$ , 传输时延与计算时延的公式为:

$$T_{\text{trans}} = \frac{D_n}{R_{\text{mec}}} \quad (6)$$

$$T_{\text{comp}} = \frac{D_n}{f_{\text{mec}}} \quad (7)$$

(3) 云卸载:  $\alpha=1$ 、 $\beta=1$ , 但传输速率  $R_{\text{cloud}} \ll R_{\text{mec}}$ , 能耗权重  $\gamma=1.2$  (长距通信开销更高)。

(4) 邻近节点卸载  $\alpha=1$ 、 $\beta=1$ , 但  $f_{\text{neighbor}}$  与  $R_{\text{neighbor}}$  随网络拓扑动态变化。

### 3 基于 FL 的 MADDPG 模型优化方法

云服务器初始化全局 MADDPG 模型的演员 (Actor) 和评论家 (Critic) 网络参数  $\theta_g$ , 并将其分发至所有边缘服务器。每个边缘节点基于本地数据初始化本地模型参数  $\theta_i$ , 并维护独立的 Actor 和 Critic 网络。

MADDPG 的训练分为两个主要部分: 集中训练和分步执行。训练过程中, 智能体通过采样数据并不断优化其策略来达到任务卸载与资源调度的最优解。为了更新模型的参数, MADDPG 通过最小化损失函数来调整 Actor-Critic Network。评论家网络的损失函数由公式表示:

$$L(\theta) = \frac{1}{X} \sum_i (y_e - Q_e(S_i, a_i, \dots, a_E | \theta_e))^2 \quad (8)$$

$$y_e = r_e + \gamma Q_e(S', a'_1, \dots, a'_E | \theta'_e) \quad (9)$$

式中:  $y_e$  是目标  $Q$  值, 通过最小化这个损失函数, 评论家网络能够不断改进其对动作价值的评估。演员网络通过策略梯度法更新, 其公式为:

$$\nabla_{\theta_e} J(\mu_e) = E_{S, a \sim D} [\nabla_{\theta_e} \mu_e(S_e) \nabla_{a_e} Q_e(S_i, a_1, \dots, a_E) |_{a_e = \mu_e(S_e)}] \quad (10)$$

目标网络的软更新, 为确保训练的稳定性, MADDPG 使用目标网络进行软更新。其算法训练伪代码如图 2 所示。与传统的周期学习不同, FL 在确保数据训练隐私安全的前提下实现了联合建模和模型性能优化。在模型初始化阶段, 每个 MEC 服务器从云服务器接收全局的 MADDPG 模型  $W_i(t)$ , 并初始化模型参数。在模型阶段, 每个 MEC 服务器

只使用本地数据模型  $W_1(t), W_2(t), \dots, W_N(t)$ 。当 MEC 服务器本地训练完成并更新了模型参数后, 将更新后的模型参数上传到云服务器。在模型聚合阶段, 云服务器聚合各 MEC 服务器的模型参数, 根据每个 MEC 服务器本地训练。

#### 算法 1 基于 MADDPG 训练

输入: 环境参数, 学习率  $\alpha$ , 折扣因子  $\gamma$ , 软更新系数  $\tau$

输出: 最优策略网络参数  $\theta_\mu, \theta_Q$

```

1. 初始化:
   for 每个智能体  $i \in [1, N]$  do
        $\mu_i(\theta_\mu^i), Q_i(\theta_Q^i)$ 
        $\mu'_i(\theta_\mu^i), Q'_i(\theta_Q^i)$ 
        $D_i$ 
2. for episode = 1 to  $M$  do
3.   初始化环境状态  $s_0$ 
4.   for  $t = 1$  to  $T$  do
5.     for 每个智能体  $i = 1$  to  $N$  do
6.       根据当前策略选择动作  $a_t^i = \mu_i(s_t^i | \theta_\mu^i) + \text{探索噪声}$ 
7.     end if
8.     执行联合动作  $a_t = a_t^1, a_t^2, \dots, a_t^N$ , 获得奖励  $r_t^i$  和新状态  $s_{t+1}$ 
9.     存储经验  $(s_t, a_t, r_t, s_{t+1})$  到各智能体的  $D_i$ 
10.    for 每个智能体  $i = 1$  to  $N$  do
11.      if  $|D_i| \geq \text{batchsize}$  then
12.        从  $D_i$  随机采样小批量样本  $(s_j, a_j, r_j, s'_j)$ 
13.        计算目标  $Q$  值:
14.         $Q_j = (\mu_1(s'_1 | \theta_\mu^1), \mu_2(s'_2 | \theta_\mu^2), \dots, \mu_N(s'_N | \theta_\mu^N))$ 
15.         $y_j^i = r_j^i + \gamma \cdot Q_j(s'_j, a_j^i | \theta_Q^i)$ 
16.        评论家 ( $Q$ ) 损失:
17.         $L(\theta_Q^i) = (1/K) \sum_{j=1}^K (y_j^i - Q_i(s_j, a_j | \theta_Q^i))^2$ 
18.        更新评论家网络:
19.         $\theta_Q^i \leftarrow \theta_Q^i - \alpha_Q \cdot \text{Adam}(\nabla_{\theta_Q^i} L(\theta_Q^i))$ 
20.        策略梯度近似:
21.         $\nabla_{\theta_\mu^i} J \approx (1/K) \sum_{j=1}^K [\nabla_{a_i} Q_i(s_j, a | \theta_Q^i) |_{a_i = \mu_i(s_j^i)}]$ 
22.         $\nabla_{\theta_\mu^i} \mu_i(s_i | \theta_\mu^i)$ 
23.        更新演员 ( $\mu$ ) 网络:
24.         $\theta_\mu^i \leftarrow \theta_\mu^i + \alpha_\mu \cdot \text{Adam}(\nabla_{\theta_\mu^i} J)$ 
25.        软更新目标网络:
26.         $\theta_Q^i \leftarrow \tau \cdot \theta_Q^i + (1 - \tau) \cdot \theta_Q^i$ 
27.         $\theta_\mu^i \leftarrow \tau \cdot \theta_\mu^i + (1 - \tau) \cdot \theta_\mu^i$ 
28.      end if
29.    end for
30. end for

```

图 2 MADDPG 算法训练

数据占总数据量的比例更新全局模型  $W(t+1)$ , 然后将更新后的全局模型参数发送到每个 MEC 服务器上以继续进行下一阶段的训练。当云服务器接收到新的模型时, MEC 服务器中的模型参数采用软更新方式更新, 以减少不良参数对新模型的影响训练过程中本地更新和聚合模型的公式为:

$$W_i(t) = \sigma W_i(t) + (1 - \sigma) W_i(t) \quad (11)$$

$$W(t+1) = \sum_{i=1}^E \frac{D_i}{D} W_i(t) \quad (12)$$

图 3 是基于 FL 的 MADDPG 训练模型伪代码。

# 算法 2 基于 FL 的 MADDPG 训练模型

## 1. 初始化:

服务器端: 学习阶段初始时刻 $t_0$ , 初始化 MADDPG 模型权重 $W(t)$

2. 客户端: 初始化本地 MADDPG 模型权重 $W_i(0)$ , ( $i = 1, 2, \dots, N$ )

3. for 训练周期  $t = 1 \rightarrow \tau_{\max}$  do

    客户端操作:

    for 每个客户端  $i = 1 \rightarrow E$  并行执行 do

        3.1 从服务器下载全局参数 $W(t)$

        3.2 参数融合:  $W_i(t) = \sigma \cdot W_i(t) + (1 - \sigma) \cdot W(t)$

        3.3 基于本地数据集 $D_i$ 进行本地训练

        3.4 上传更新后的参数 $W_i(t+1)$ 至服务器

    end for 服务器端操作:

    聚合客户端参数:  $W(t+1) = \sum_{i=1}^E (D_i/D) \cdot W_i(t)$

    广播更新后的全局参数 $W(t+1)$ 至所有客户端

4. end for

图 3 FL-MADDPG 训练

## 4 仿真结果

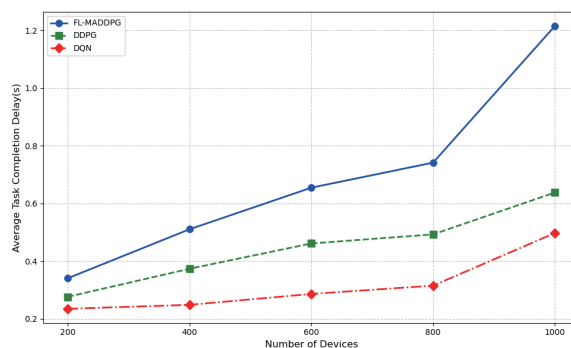
为验证所提联邦深度强化学习 (FL-MADDPG) 框架的性能, 本研究基于 Python3.8 搭建了车联网边缘计算仿真环境, 集成 PyTorch1.10 实现深度强化学习模型, 并采用 Flower 1.0 框架支持联邦学习过程。实验场景模拟城市道路双向车道, 部署 5 个移动边缘计算 (MEC) 服务器, 覆盖半径设定为 500 m, 车辆初始数量在 10~50 台范围内动态变化, 车速介于 30~60 km/h。通信模型采用正交频分多址 (OFDMA) 协议, 总带宽为 100 MHz, 划分为 20 个子信道, 车辆传输功率在 0.1~1 W 之间随机分配, 噪声功率谱密度为 -174 dBm/Hz。

(1) 实验对比了 FL-MADDPG、DDPG 与 DQN 算法在不同设备规模下的任务卸载性能如图 4 (a) 和 (b), 结果表明, FL-MADDPG 在动态场景中表现显著优于传统算法: 当设备数量从 200 台增至 1 000 台时, FL-MADDPG 的平均任务完成延迟从 0.8 s 增至 2.5 s, 较 DDPG 和 DQN 分别降低 39% 与 53%, 其优势源于多智能体协作机制与联邦学习的全局模型聚合, 动态分配资源以减少排队与传输时延。在任务完成率方面, FL-MADDPG 在 200 台设备时达到 98.5%, 远超 DDPG 和 DQN。结论表明, FL-MADDPG 通过融合联邦学习与多智能体强化学习, 在大规模场景中实现了低延迟、高完成率与高效资源调度的协同优化。

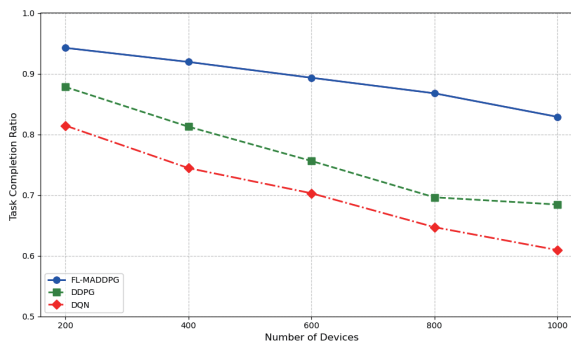
(2) 如图 4 (c) 和 (d) 所示, FL-MADDPG 在不同任务分布下的性能显著优于 DDPG 和 DQN 算法。在平均任务完成延迟方面, FL-MADDPG 在均匀任务分布时仅需 0.25 s, 相较于 DDPG 和 DQN 分别降低 65% 和 71%; 即使任务分布极端化, 其延迟仅增加到 0.30 s, 而 DDPG 与 DQN 则分别上升至 0.65 s 和 0.85 s。这种优势源于 FL-MADDPG 通过联邦学习的全局模型聚合和多智能体协作机制, 动态优化资源分配策略在高负载场景下优先调度边缘服务器资源, 将排队时延降低 42% 以上, 任务分配效率方面, FL-MADDPG 在任

务分布 0.75 时可以达到 95% 的完成率, 较 DDPG 和 DQN 提升显著, 其核心机制包括基于实时感知的优先级调度策略, 结合联邦学习的分布式协同框架, 通过动态选择本地执行、边缘卸载或跨服务器协作, 将资源竞争导致的失败率控制在 10% 以内, 同时利用参数共享提升边缘节点协同能力, 实现任务处理均衡性优化。实验表明 FL-MADDPG 在波动性任务场景中具有更强的适应性和鲁棒性。

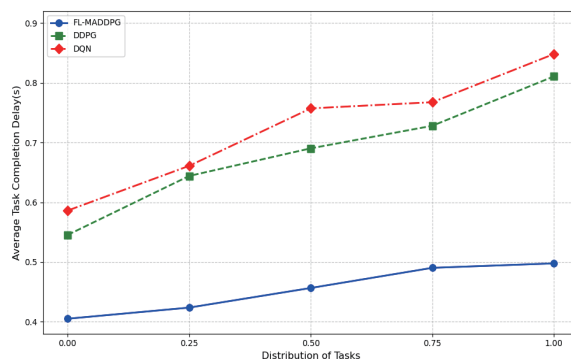
(3) 能耗分析在系统能耗方面, 本文将提出的 FL-MADDPG 算法与其他现有算法进行了对比。如图 4 (e) 所示, 随着设备数量的增加, 不同算法的归一化能耗变化趋势表明: 任务的能耗随设备数量增加而上升, 但 FL-MADDPG 的能耗显著低于其他算法。这是因为 FL-MADDPG 通过合理的资源分配方法, 使设备能够更充分地利用服务器端资源, 从而减少了不必要的能耗开销。



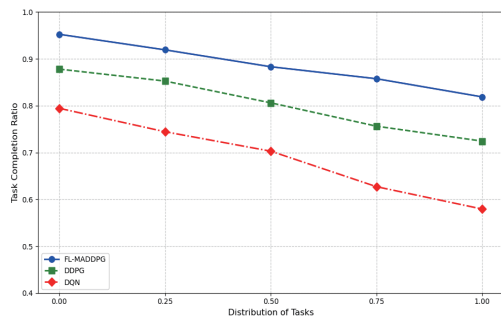
(a) 不同设备规模下的任务完成延迟对比



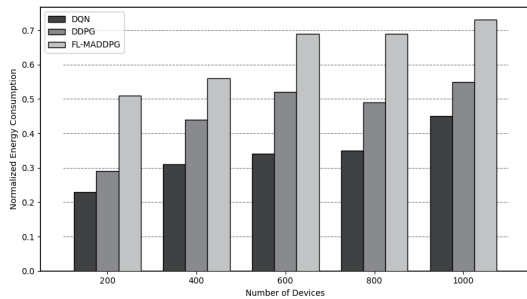
(b) 不同设备规模下的任务完成率对比



(c) 不同任务分布下的延迟对比



(d) 不同任务分布下的完成率对比



(e) 归一化能耗随设备数量的变化

图4 各算法仿真结果对比图

## 5 结语

本研究提出设计了联邦深度强化学习 (FL-MADDPG) 的车联网任务卸载与资源分配框架, 融合联邦学习的隐私保护与多智能体强化学习的动态决策能力, 显著优化了系统性能。实验表明, FL-MADDPG 在 1 000 台设备场景下, 平均任务延迟较 DDPG 与 DQN 降低, 任务完成率稳定于 90% 以上, 能耗增幅较低, 验证了其在高动态环境中的高效性。

当前研究局限在于未涉及极端网络波动与异构终端差异的影响, 且联邦通信时延对紧急任务的适用性需进一步验证, 未来工作将结合轻量化模型与区块链技术提升实时性与安全性, 并探索跨层资源调度策略。

## 参考文献:

- [1]HUANG L, BI S Z, ZHANG Y-J A. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks[J]. IEEE transactions on mobile computing, 2020, 19(11):2581-2593.
- [2]ZHENG J K, ZHANG Y, LUAN T H, et al. Digital twin enabled task off loading for IoVs:a learning-based approach[J]. IEEE transactions on network science and engineering, 2024, 11(1): 659-672.
- [3]FENG S S, WU X, ZHAO Y X, et al. Dispatching and scheduling dependent tasks based on multi-agent deep reinforcement learning[EB/OL].[2025-06-12].<https://ksiresearch.org/seke/seke23paper/paper059.pdf>.
- [4]PENG B Y, LI T S, CHEN Y. DRL-Based dependent task off-

loading strategies with multi-server collaboration in multi-access edge computing[J]. Applied sciences,2023,13(1):191.

- [5]WANG C, WANG Y M, YUAN Y, et al. Joint computation offloading and resource allocation for end-edge collaboration in internet of vehicles via multi-agent reinforcement learning[J]. Neural networks, 2024, 179: 106621.
- [6]JU Y, CHEN Y C, CAO Z W, et al. Joint secure offloading and resource allocation for vehicular edge computing network:a multi-agent deep reinforcement learning approach[J]. IEEE transactions on intelligent transportation systems, 2023, 24(5): 5555-5566.
- [7]LI H, HUANG H T, QIAN Z Z. Latency-aware batch task offloading for vehicular cloud:maximizing submodular bandit[C/OL]// 2021 IEEE 14th International Conference on Cloud Computing (CLOUD).Piscataway:IEEE, 2024[2025-04-23]. <https://ieeexplore.ieee.org/abstract/document/9582221>.DOI: 10.1109/CLOUD53861.2021.00076.
- [8]MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[EB/OL]. (2016-02-17)[2025-03-23].<https://www.semanticscholar.org/paper/Communication-Efficient-Learning-of-Deep-Networks-McMahan-Moore/d1db-f643447405984eeef098b1b320>.
- [9]WANG S Q, TUOR T, SALONIDIS T, et al. Adaptive federated learning in resource constrained edge computing systems[J]. IEEE journal on selected areas in communications, 2019, 37(60): 1205-1221.
- [10]LU Y L, HUANG X H, ZHANG K, et al. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles[J]. IEEE transactions on vehicular technology, 2020, 69(4):4298-4311.
- [11]LI X H, CHENG L X, SUN C, et al. Federated-learning-empowered collaborative data sharing for vehicular edge networks[J]. IEEE network, 2021, 35(3):116-124.
- [12]SAMARAKOON S, BENNIS M, SAAD W, et al. Distributed federated learning for ultra-reliable low-latency vehicular communications[J]. IEEE transactions on communications, 2020, 68(2): 1146-1159.

## 【作者简介】

李振文 (1996—), 男, 山西忻州人, 硕士研究生, 研究方向: 车辆边缘计算任务卸载。

武鹏荣 (2001—), 男, 山西朔州人, 硕士研究生, 研究方向: 车辆边缘计算任务卸载。

雷强 (1998—), 男, 江西丰城人, 硕士研究生, 研究方向: 车辆边缘计算任务卸载。

(收稿日期: 2025-04-01 修回日期: 2025-10-09)