车联网中活跃度感知的云边协同内容缓存方法

胡前灵[1]，王宇翱[1]，孙洪波[1,2]，郭永安[1]

（1. 智能信息处理与通信技术省高校重点实验室，江苏南京，210003；
2. 边缘智能研究院南京有限公司，江苏南京，210003）

摘　要：针对边缘缓存中路边单元（Roadside unit, RSU）存储空间有限导致用户体验质量（Quality of experience, QoE）不均衡的问题，提出一种车辆用户（Vehicle user, VU）活跃度感知的云边协同数据缓存方法，旨在提升内容缓存服务的公平性并保障 QoE。构建车联网三层云边协同缓存系统模型，通过联合优化服务方式和缓存替换，最大化系统净收益，保障内容交付的及时性和有效性。基于活跃度差异对 VU 进行聚类，以获取个性化内容预测结果；设计内容请求预测模型，利用联邦学习框架实现全局模型更新。将系统长期收益最大化问题转化为多智能体马尔可夫决策过程（Markov decision process, MDP），并采用多智能体深度确定性策略梯度（Multi-agent deep deterministic policy gradient, MADDPG）算法得到最优决策。仿真结果验证了解决方案的有效性，与其他基准方法相比，提高了平均系统收益和缓存命中率，同时有效保障系统公平性。

关键词：车联网；协作缓存；内容预测；多智能体深度强化学习；联邦学习

中图分类号：TN929.52　　**文献标志码：**A　　文章编号：

Activity Awareness Collaborative Cloud-Edge Content Caching Method in Internet of Vehicles

HU Qianling [1], WANG Yuao[1], SUN Hongbo[1,2], GUO Yongan[1],

(1. Provincial Key Laboratory of Intelligent Information Processing and Communication Technology, Nanjing, Jiangsu, 210003, China
2. Edge Intelligence Research Institute Nanjing Co., LTD., Nanjing, Jiangsu, 210003, China)

Abstract: Aiming at the problem of unbalanced quality of experience (QoE) caused by limited storage space of roadside unit (RSU) in edge caching, a vehicle user (VU) activity-aware cloud-edge collaborative data caching method is proposed to improve the fairness of content caching services and ensure QoE. The three-tier cloud-edge collaborative caching system model of the Internet of Vehicles is constructed to maximize the net revenue of the system and ensure the timeliness and effectiveness of content delivery by jointly optimizing service mode and cache replacement. VU is clustered based on different activity to obtain personalized content prediction results. The content request prediction model is designed, and the global model update is realized by using the federated learning framework. The problem of maximizing the long-term profit of the system is transformed into a multi-agent markov decision process (MDP), and the multi-agent deep deterministic policy gradient (MADDPG) algorithm is used to obtain the optimal decision. The simulation results verify the effectiveness of the solution. Compared with other baseline methods, the average system revenue and cache hit rate are improved, and the system fairness is effectively ensured.

Keywords: Internet of vehicles, cooperative caching, content prediction, multiagent deep reinforcement learning, federated learning

---

收稿日期：2025−08−18；**修回日期：**2025−09−30

基金项目：江苏省前沿引领技术基础研究专项项目（BK20202001）；江苏省重点研发计划重点项目（BE2023025）；江苏省研究生科研与实践创新计划项目（SJCX25_0299，SJCX24_0285）。

引言

随着物联网技术的快速发展，网联汽车数量呈爆发式增长。传统云中心由于部署位置集中且与车辆用户（Vehicle user, VU）距离较远，难以提供低时延的内容交付。此外，其集中式数据交互方式可能导致核心网络拥塞[1-2]，对车联网发展带来严峻挑战。边缘缓存技术通过将热门内容预存于部署在网络边缘的路边单元（Roadside unit, RSU），使 VU 能够直接从 RSU 获取请求内容，从而有效降低交付延迟并减轻核心网络负载[3-4]。然而，与云服务器相比，RSU 的缓存容量存在显著限制[5]，难以满足全部 VU 的内容请求。协同缓存技术通过 VU、RSU 与云服务器的资源协作和数据共享，实现分布式存储和调度，能够扩展 VU 的内容可获取范围，提升请求满足率。但这种方式在跨节点数据传输过程中会引入额外时延[6]。因此，需要设计高效的缓存策略，以支持 VU、RSU 和云服务器之间的无缝协作。基于规则的传统缓存策略，如最近最少使用（Least recently used, LRU）[7]和最不常用（Least frequently used, LFU）[8]虽被广泛使用，但完全依赖于历史请求信息，难以适应内容请求和网络的动态性。当前，针对云边协同的数据缓存，研究重点普遍聚焦于精准内容请求预测与缓存决策制定两方面。

内容流行度预测是协作缓存策略的核心前提。通过分析各 VU 差异化的内容请求偏好，可捕捉其需求趋势，从而为有限缓存资源的合理分配提供依据。机器学习方法能够从历史请求数据中提取潜在特征[9-11]，并已广泛应用在热门内容预测。然而，传统机器学习需要将数据集中上传到中心服务器进行处理，难以保障用户隐私。联邦学习作为一种分布式训练框架，无需共享原始数据即可通过聚合来自 VU 的本地模型参数更新全局预测模型[12-14]，有效缓解了 VU 信息泄露风险。已有研究探索了联邦学习在车联网内容预测中的应用，例如，文献[12]在车辆部署上下文感知对抗性自动编码器（Context-aware adversarial autoencoder, C-AAE）模型，并采用联邦学习训练模型用于预测内容受欢迎程度。文献[13]提出基于多级联邦学习的协同缓存算法以兼顾数据安全和缓存效率。文献[14]结合联邦学习、移动性预测和一致性哈希实现热门内容确定。文献[15]设计了去中心化、个性化的虚拟现实服务缓存算法。然而，这些方法普遍忽略了 VU 活跃度差异对预测结果的影响。

活跃度通常由 VU 的内容请求频率和有效活跃时长（在 RSU 覆盖范围内的停留时间）等行为特征衡量。在车联网中，不同 VU 活跃度的异质性导致其内容请求呈现显著差异：高活跃度 VU 往往具有更稳定的内容偏好和可预测的请求模式，而低活跃度 VU 的请求更具随机性。此外，RSU 通常优先缓存访问频率较高的热门内容[16]，这可能使低活跃度 VU 的部分请求被忽略，导致缓存失效和响应延迟等问题。在缓存策略制定中，综合考虑 VU 的活跃度与内容偏好，有助于提高预测准确性，为后续缓存决策提供更可靠的依据，从而优化整体缓存命中率，并减少不同 VU 间的服务体验质量（Quality of experience, QoE）差异。

尽管 RSU 能够通过预测缓存部分热门内容，但受限于缓存容量，难以覆盖全部潜在需求。为此，需要面向动态环境和时序特性的高效协同缓存策略，以在多个 RSU 间合理分配缓存任务，提升系统的整体性能与适应性[17]。协同缓存涉及到云和边缘在动态环境下的交互，需要联合考虑通信和存储资源，往往难以求解。深度强化学习（Deep reinforcement learning, DRL）可通过马尔可夫决策过程（Markov decision process, MDP）建模车联网的动态特征，凭借其环境认知和长期决策能力[18-19]，已被广泛应用于车载缓存优化。已有研究如文献[20]基于深度确定性策略梯度（Deep deterministic policy gradient, DDPG）优化缓存替换策略。文献[21]基于深度 Q 网络（Deep Q-learning network, DQN）联合优化缓存位置和内容放置，提

高 RSU 的内容可用性。文献[22]基于双重 DQN（Deep double Q-learning network, DDQN）设计了缓存替换机制，缓解了 Q 值估计偏差，提高了学习稳定性与系统收益。但随着 VU 数量和内容规模的不断增长，决策空间呈指数级扩展，集中式单智能体方法在高维决策空间中的学习效率与泛化能力面临挑战。

为了解决高维决策问题，多智能体深度学习（Multi-agent deep reinforcement learning, MADRL）成为新的研究趋势，使多个独立智能体分布式交互与协作决策。文献[23]采用多智能体 DDPG（Multi-agent deep deterministic policy gradient, MADDPG）协调跨基站的缓存放置。文献[24]集成了 DQN 和柔性动作-评价（Soft actor-critic, SAC）构建分层缓存与卸载框架，以实现云边跨层决策。文献[25]采用平均域 MADRL 缓解边缘缓存中的冷启动问题。文献[26]提出了一种基于 MADRL 的空地协同缓存算法，以最大化动态内容请求下全局缓存命中率。文献[27]通过分层博弈论模型和 DRL 结合来提升缓存策略稳定性。尽管已有研究在内容预测或缓存决策方面取得一定进展，但尚未形成能够协同整合两者并兼顾 VU 活跃度差异的统一框架。

为了解决上述问题，本文提出了一种融合内容流行度预测与 DRL 缓存决策的分层缓存策略体系。在预测阶段引入 VU 活跃度特征，以提升预测精度并兼顾不同 VU 的服务公平性；在决策阶段采用基于预测结果的多智能体 DRL 策略，实现云、边、端的协同缓存优化，提升系统自适应能力、缓存效率和整体服务性能。

（1）构建了由 VU、RSU 及云服务器构成的三层车联网云边协同架构，并建立系统模型。在该模型中，VU 的内容请求可由本地 RSU、相邻 RSU 或云服务器响应。通过引入内容传输成本、公平服务成本与系统收益等指标，将系统净收益最大化问题进行公式化建模，以优化决策并保障内容交付的及时性和有效性。

（2）提出活跃度联合量化指标，综合衡量 VU 的内容请求频率与停留时间；设计基于活跃度的内容请求预测算法，并构建三层联邦训练框架实现全局预测模型更新。训练完成的预测模型针对不同活跃度 VU 集群生成个性化预测结果，再通过加权融合获得最终预测结果，为缓存决策提供可靠依据。

（3）针对协同缓存场景，构建多智能体 MDP 模型，并引入 MADDPG 算法进行缓存决策优化，通过最大化系统总收益求解全局最优的协同缓存策略，从而提升云边协同缓存场景下的缓存效率与服务质量。

1.系统模型与问题建立

1.1 系统整体模型

如图 1 所示，所提出的车联网协同缓存系统由一个宏基站（Macro base station, MBS），M 个部署边缘服务器的 RSU 以及U 个 VU 组成。宏基站内部署了云服务器，有充足的容量来存储 VU 需要的各类内容，内容的集合定义为$\mathrm{C}=\{1,2,...,C\}$，其中每个内容$c\in\mathrm{C}$的大小为S_c。RSU 的集合被定义为$\mathrm{M}=\{1,2,...,M\}$，每个 RSU 的覆盖区域不发生重叠，可从宏基站处通过光纤缓存有限数量的内容，进一步为其覆盖范围内的 VU 提供内容服务，设 RSU$m\in\mathrm{M}$ 的存储容量为S_m。VU 的集合被定义为$\mathrm{U}=\{1,2,...,U\}$，通过无线链路连接到其本地 RSU。

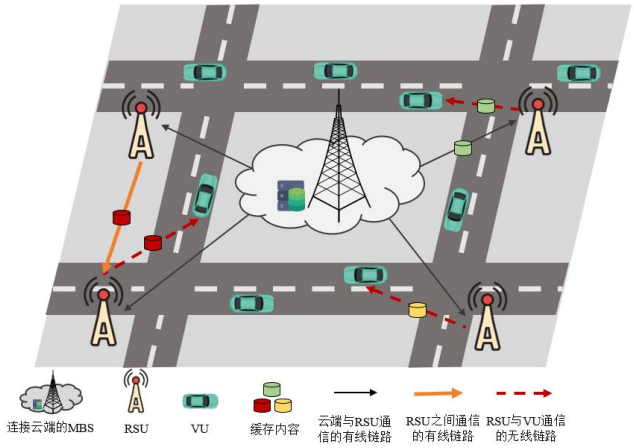

图 1 车联网云边协同系统模型

Fig.1 System model of cloud-edge collaborative in the internet of vehicles

在缓存系统中，服务时间被划分为T段，集合定义为$\mathrm{T}=\{1,2,...,T\}$。由于各 RSU 的内容缓存情况不同，VU 请求的内容可以由本地 RSU、相邻 RSU 或 MBS 提供。令$\rho_{u,c}^{t}=1$表示 VUu在时隙$t\in\mathrm{T}$中请求内容c，否则$\rho_{u,c}^{t}=0$。本文考虑一个现实的场景，其中 VU 的内容请求是随机的，并且大致遵循 Zipf 分布[28]。

在时隙t内，RSUm的覆盖区域中存在U_m^t辆 VU，采用集合$\mathrm{U}_m^t\in\left\{1,2,\cdots,U_m^t\right\}$表示。此外，假设每个 VU 在时隙内保持稳定的移动特性，并且不同 VU 的速度遵循独立同分布（i.i.d.）分布。设 VU$u\in\mathrm{U}_m^t$在 RSUm覆盖区域内的速度为$v_{m,u}^t$，其概率密度函数表示为

$$f_v=\begin{cases}\dfrac{e^{-\frac{1}{2\gamma^2}(v_{m,u}^t-\mu)^2}}{\gamma\sqrt{2\pi}\left(erf\left(\dfrac{v_h-\mu}{\sqrt{2}\gamma}\right)-erf\left(\dfrac{v_l-\mu}{\sqrt{2}\gamma}\right)\right)}, & v_l\le v_{m,u}^t\le v_h\\ 0, & \text{othersize}\end{cases}\tag{1}$$

式中：v_h和v_l分别是城市交通规定的最大和最小速度，$erf\left(\dfrac{v_{m,u}^t-\mu}{\sqrt{2}\gamma}\right)$是在均值$\mu$和方差$\gamma^2$下的高斯误差函数。

设 VUu在 RSUm覆盖范围内的移动距离为$Z_{m,u}^t$，表示为

$$Z_{m,u}^t=Z_{m,u}^{t-1}+v_{m,u}^t\Delta t\tag{2}$$

对于每个 VUu，可以根据其速度计算出在本地 RSU 覆盖范围内的剩余行驶时间为

$$T_{m,u}^{t,cov}=(Z_d-Z_{m,u}^t)/v_{m,u}^t\tag{3}$$

式中：Z_d是 RSU 覆盖范围的直径。

1.2 内容传输成本模型

如图 1 所示，当 VU 向本地 RSU 发出内容请求时，有三种方式来提供内容c。

（1）本地 RSU 提供内容服务

最理想的情况是 VU 连接的本地 RSU 提前缓存该内容，可以直接通过无线链路传输。在考虑的网络中，RSU 采用**正交频分复用（Orthogonal frequency division multiplexing，OFDM）技术**与覆盖范围内的 VU

进行无干扰的通信。则 VU u 时隙 t 的信道增益建模为

$$G_{m,u}^{t}=\alpha_{u}^{t}(dis(m,u))l_{u}^{t},u\in \mathrm{U}_{m}^{t} \tag{4}$$

式中：$dis(m,u)$ 是 RSU m 与 VU u 之间的传输距离，$\alpha_{u}^{t}(dis(m,u)$ 是大规模衰落效应，包括路径损耗和阴影。l_{u}^{t} 是小尺度衰落效应，假设其单位均值呈指数分布[21]。RSU m 和 VU u 之间的传输速率 $r_{m,u}^{t}$ 表示为

$$r_{m,u}^{t}=\frac{W}{U_{m}^{t}}\log_{2}(1+\frac{P_{m}G_{m,u}^{t}}{\sigma^{2}}) \tag{5}$$

式中：P_{m} 是 RSU m 的发射功率，σ^{2} 是噪声功率，W 是 RSU 的带宽。

则 RSU m 将内容 c 传输到 VU u 的成本表示为

$$d_{m,u,c}^{t}=\alpha_{1}\frac{s_{c}}{r_{m,u}^{t}} \tag{6}$$

式中：α_{1} 表示利用无线链路传输内容的单位成本。

（2）相邻 RSU 提供内容服务

如果本地 RSU 无法提供内容 c，但相邻 RSU n 缓存了该内容，则可以通过光纤传输给本地 RSU，然后再通过无线链路下发，此时的内容传输成本表示为

$$d_{n,u,c}^{t}=\alpha_{1}\frac{s_{c}}{r_{m,u}^{t}}+\alpha_{2}\frac{s_{c}}{r_{m,n}^{t}} \tag{7}$$

式中：$r_{m,n}^{t}$ 是 RSU m 和相邻 RSU n 之间的有线链路传输速率，α_{2} 是利用有线链路传输内容的单位成本。

（3）MBS 提供内容服务

如果内容 c 既没有缓存在 RSU m 中也没有缓存在相邻 RSU n 中，则本地 RSU 需要向 MBS 请求内容 c，然后传输到 VU u，内容传输成本表示为

$$d_{b,u,c}^{t}=\alpha_{1}\frac{s_{c}}{r_{m,u}^{t}}+\alpha_{2}\frac{s_{c}}{r_{m,b}^{t}} \tag{8}$$

式中：$r_{m,b}^{t}$ 是 RSU m 和 MSB b 之间的有线链路传输速率。

定义集合 $g_{m,c}^{t}=(g_{u,c}^{m,t},g_{u,c}^{n,t},g_{u,c}^{b,t}),u\in \mathrm{U}_{m}^{t},c\in \mathrm{C}$ 表示满足 VU u 内容 c 请求的服务方式，其中 $g_{u,c}^{m,t},g_{u,c}^{n,t},g_{u,c}^{b,t}\in\{0,1\}$，分别表示 VU u 的内容请求由本地 RSU m、相邻 RSU n 或 MBS b 提供。将 RSU m 的缓存服务方式表示为 $\boldsymbol{g}_{m}^{t}=\{g_{m,1}^{t},\cdots,g_{m,c}^{t}\cdots,g_{m,C}^{t}\}$。且服务方式的选择满足约束 $g_{u,c}^{m,t}+g_{u,c}^{n,t}+g_{u,c}^{b,t}=1$。

由于系统支持三种内容服务模式，则每个 RSU 的内容传输成本表示为

$$D_{m}^{t}=\sum_{u\in \mathrm{U}_{m}^{t}}\sum_{c\in \mathrm{C}}\rho_{u,c}^{t}(g_{u,c}^{m,t}d_{m,u,c}^{t}+g_{u,f}^{n,t}d_{n,u,c}^{t}+g_{u,f}^{b,t}d_{b,u,c}^{t}) \tag{9}$$

此外，由于 RSU m 的存储空间有限，需要考虑缓存或替换的内容。将 RSU m 的缓存动作表示为 $\boldsymbol{c}_{m}^{t}=\{c_{m,0}^{t}\}\bigcup\{c_{m,1}^{t},\cdots,c_{m,c}^{t}\cdots,c_{m,C}^{t}\}$，当 $c_{m,c}^{t}=1$ 时，表示 RSU m 选择将内容 c 缓存下来，否则 $c_{m,c}^{t}=0$。特别地，如果 $c=0$ 并且 $c_{m,0}^{t}=1$，则表示时隙 t 内 RSU m 不替换任何缓存内容。

1.3 公平服务成本

每个本地 RSU 的目标是可以直接向其覆盖范围内的 VU 直接提供内容，而无需向相邻 RSU 或者 MBS 请求，这可以显著降低内容传输成本。针对 VU u，定义其内容请求直接在本地 RSU 处满足为缓存命中，

则时隙 t 处的内容缓存命中数量表示为

$$h_u^t = \sum_{c=1}^{C} \rho_{u,c}^t g_{u,c}^{m,t} \tag{10}$$

然而，如果在 RSU m 处的缓存决策仅仅是为了最大化全局缓存命中总数 $\sum_{u=1}^{U_m^t} h_u^t$，那么缓存内容分配将显著偏向于高活跃度的 VUs，来自低活跃度 VUs 的请求需要从远程 MBS 获取，从而导致额外的传输延迟以及显著的 QoE 下降。为了保证 VU 体验的一致性，应该均衡缓存以尽可能满足大部分 VU 的内容请求。因此，本文引入 Jain's 公平指数来衡量 RSU m 的服务公平性[29-30]，表示为

$$J_m^t = \frac{(\sum_{u=1}^{U_m^t} h_u^t)^2}{U_m^t (\sum_{u=1}^{U_m^t} (h_u^t)^2)} \tag{11}$$

式中：U_m^t 是 RSU m 在时隙 t 覆盖范围内的 VU 个数。根据 Cauchy-Buniakowsky-Schwarz 不等式，得到 $0 \le J_m^t \le 1$。表明 VU 之间的缓存命中数差异越大，公平性指数将越小。而公平性指数越大，意味着 RSU 的缓存服务越公平。为了平衡 RSU m 的内容传输成本和公平服务成本，总成本表示为

$$K_m^t = \frac{D_m^t}{J_m^t} \tag{12}$$

1.4 收益模型

VU u 发出请求之后，获得内容的等待时间表示为

$$T_{m,u,c}^t = \sum_{c \in C} \rho_{u,c}^t [\frac{s_c}{r_{m,u}^t}(g_{u,c}^{m,t} + g_{u,c}^{n,t} + g_{u,c}^{b,t}) + g_{u,c}^{n,t} \frac{s_c}{r_{m,n}^t} + g_{u,c}^{b,t} \frac{s_c}{r_{m,b}^t}] \tag{13}$$

作为内容服务提供方，RSU 的收益与内容传输延迟成反比，其收益函数表示为

$$C_{m,u,c}^t = e^{-\tau T_{m,u,c}^t} Y_c \tag{14}$$

式中：τ 表示衰减因子，Y_c 表示内容 c 的价值，这直接决定了 RSUs 的收益。

每个内容请求都有一定的等待时间限制，定义索引 $IX_{m,u,c}^t$ 判断缓存请求是否在其限制时间 T_c 之内得到满足，表示为

$$IX_{m,u,c}^t = \begin{cases} 1, & T_{m,u,c}^t \le T_c \\ -1, & T_{m,u,c}^t > T_c \end{cases} \tag{15}$$

如果内容成功传输，则索引 $IX_{m,u,c}^t = 1$；如果失败，则索引 $IX_{m,u,c}^t = -1$。则 RSU m 的收益表示为

$$E_m^t = \sum_{u \in U_m^t} \sum_{c \in C} \rho_{u,c}^t IX_{m,u,c}^t C_{m,u,c}^t \tag{16}$$

1.5 问题建立

RSU 的共同目标是在降低成本的同时提高服务收益，为了体现整体服务的公平性和缓存决策的独立性，将单个 RSU 的净收益建模为其独立服务收益与系统平均成本的差值。系统平均成本表示为

$$K^t = \frac{\sum_{m=1}^{M} K_m^t}{M} \tag{17}$$

通过优化缓存服务方式 $\boldsymbol{g}_m^t$ 和缓存动作 $\boldsymbol{c}_m^t$，最大化每个 RSU 的净收益 $E_m^t - K^t$，则优化问题表示为

$$\begin{aligned}&\max_{\boldsymbol{c}_m^t,\boldsymbol{g}_m^t}\sum_t(E_m^t-K^t)\\&\text{s.t. C1}:\sum_{u\in U_m^t}w_{m,u}^t\le W_m\\&\quad\text{C2}:\sum_{c\in C}a_{m,c}^t s_c\le S_m\\&\quad\text{C3}:g_{u,c}^{m,t}+g_{u,c}^{n,t}+g_{u,c}^{b,t}=1,u\in U_m^t\end{aligned}\tag{18}$$

式中：W_m是 RSU m 的总带宽，S_m是 RSU m 的缓存容量。约束 C1 表示 RSU m 的总带宽限制。约束 C2 表明了 RSU 缓存内容时需要考虑存储容量限制。约束 C3 定义了只能在一个地方检索到 VU 请求的内容。综合考虑系统成本和收益的优化问题由整数变量$\boldsymbol{c}_m^t,\boldsymbol{g}_m^t$组成。此外，VU、RSU 和 MBS 的状态是高度动态和相互关联的，因此各种不确定变量可以显著影响内容交付时间。此外，由于缓存决策的解空间很大，因此获得最优策略是一个挑战。

2.预测算法设计

优化问题的核心目标在于系统收益最大化，而这一目标的实现离不开准确的内容预测。内容预测通过提前识别高缓存概率的内容，并将其预加载至 RSU，可以为决策提供更好的参考，显著提升 VU 的缓存命中率，从而直接推动系统收益的增长。进一步的，引入联邦学习对预测模型进行训练，能够充分利用多 VU 的分布式数据，增强模型对内容偏好差异性的刻画能力，在保持数据隐私的同时提升预测精度，进而做出更优的缓存决策，最终使系统收益进一步提高。

本文提出了基于活跃度的内容请求预测（Content request prediction based on activity, CRP-BA）算法，并设计了三层联邦学习框架以实现预测模型的分布式训练与全局聚合。将 VU 根据活跃度水平分成高中低三个集群，训练好的预测模型可以从 VU 数据中提取潜在特征和内容受欢迎程度评级，识别每个集群中最受欢迎的内容。通过给不同活跃度的 VU 集群设置权重以保障服务公平性，最终实现对内容请求的精准预测。

2.1 基于联邦学习的预测模型训练

本文采用自编码器（Autoencoder, AE）作为预测模型，并通过联邦学习框架对其进行训练，旨在充分挖掘 VU 的个体特征，增强全局模型泛化性；训练完成后，所得到的全局模型将被应用于内容流行度的预测任务中。

基于联邦学习框架的模型训练过程如图 2 所示。RSU 会选择本轮参与训练的 VU 并向其发送上一轮的全局模型，VU 利用本地数据进行训练。在本地训练之后，VU 将局部模型上传至本地 RSU，本地 RSU 聚合模型参数以更新边缘局部模型。更新后的边缘局部模型被 RSU 发至 MBS 聚合以得到最终的全局模型。具体的训练过程如下：

（1）VU 选择和模型发布。在此阶段，本地 RSU 首先根据 VU 的行驶状态，计算其在覆盖范围内的行驶时间，并判断是否能成功参与本轮训练[21]。当 VU 的行驶时间$T_{m,u}^{\text{cov},r}$超过训练时间$T_{m,u}^{tra,r}$时，方可选中参与本轮训练。

（2）模型训练。VU 接收到云端下发的全局模型ω^{r-1}后，根据全局参数和局部偏好数据进行模型训练。VU u 的目标是找到使损失值最小的最优模型参数，即求解

$$\omega^{*}=\arg\min F(\omega^{r}) \tag{19}$$

式中：$F(\cdot)$为全局模型的损失函数。不能按时完成训练的 VU 将不能参加下一轮训练。

（3）模型聚合。每个 RSU 根据 VU 用于模型训练过程的数据量，确定每个 VU 的模型权重并进行边缘聚合，边缘聚合模型参数表示为

$$\omega_{m}^{r}=\sum_{m=1}^{M}\frac{N_{u}^{r}}{N_{m}^{r}}\omega_{u}^{r,E} \tag{20}$$

式中：N_{u}^{r}是被选中参与训练 VU 的本地数据量，N_{m}^{r}是 RSU m 覆盖区域内的训练数据总量，$\omega_{u}^{r,E}$为 VU u 完成本地训练上传的局部模型。每个 RSU 将聚合后得到的边缘局部模型ω_{m}^{r}发送至 MBS 后，在 MBS 聚合所有 RSU 的边缘局部模型。因此，全局模型参数表示为

$$\omega^{r}=\sum_{m=1}^{M}\frac{N_{m}^{r}}{N^{r}}\omega_{m}^{r} \tag{21}$$

式中：$N^{r}=\sum_{m\in M}N_{m}^{r}$是总数据量。

聚合后的全局模型ω^{r}将被下发至所有 RSU 和 VU，作为下一全局轮次的初始模型。

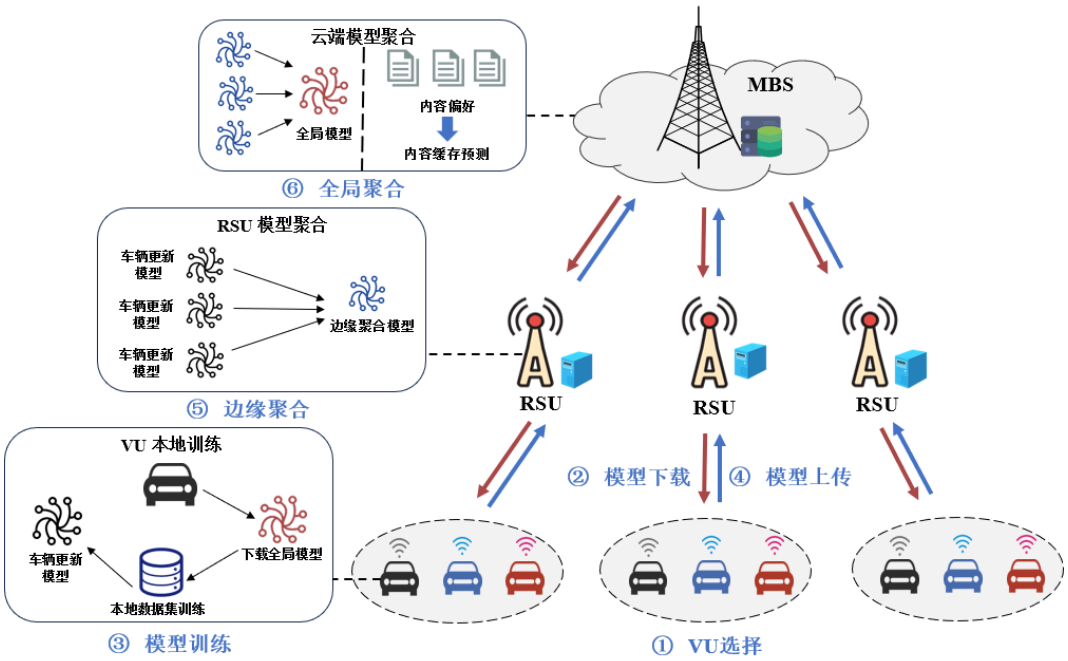

图 2 基于联邦学习的预测模型训练

Fig.2 Training of predictive models based on federated learning

2.2 基于活跃度的内容请求预测算法

在完成基于联邦学习的模型训练后，本文利用训练得到的 AE 模型进行内容流行度预测。**然而，由于 VU 活跃度存在显著差异，高活跃度车辆的请求往往在预测结果中占据优势。若在预测过程中忽略这一差异，则结果会过度偏向高活跃度群体，导致低活跃度车辆的需求被长期忽视，不仅削弱整体预测的公平性，还可能使部分紧急或突发的请求得不到及时响应，从而降低系统的服务质量。为缓解该问题，本文引入基于活跃度的聚类方法，对 VU 进行合理划分，并在预测阶段按照各集群的权重对结果进行融合。该方法既**

避免了预测结果被少数高活跃度群体主导，又兼顾了不同类型 VU 的内容偏好，在提升预测准确性的同时保障了公平性与服务可靠性。

本文具体的预测流程如图 3 所示。将每个 RSU 覆盖范围内的 VU 划分为高、中、低三个活跃度集群，以区分各个 VU 的特征差异。AE 被用来可以提取 VU 潜在的特征信息和内容评级，进一步地，选出每个集群中内容流行度最高的内容。由于 VU 的请求活跃度可能会随时间或环境而变化，本文设置了权重分配策略以对不同 VU 集群预测的流行内容进行差异化选取，这有助于增强缓存的公平性。图 3 示出了单个 RSU 的内容预测过程，包括以下步骤。

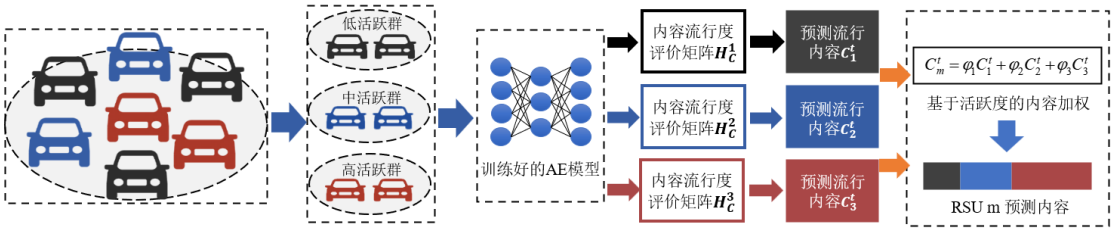

图 3 基于活跃度的内容流行度预测

Fig.3 Content popularity prediction based on activity level

（1）VU 聚类。基于 RSU m 覆盖范围内 VU 的相似特性（包括历史请求信息、速度、位置等）将所有 VU 分为高、中、低三个活跃度集群，**每个 VU 的**活跃度被定义为 $A_u^t=(\overline{CF}_u^t,\overline{T}_u^{t,\mathrm{cov}})$，其中 $\overline{CF}_u^t$ 和 $\overline{T}_u^{t,\mathrm{cov}}$ 分别是归一化后的内容请求频率和停留时间。首先，初始化高活跃度、中活跃度、低活跃度 3 个集群中心为 $v_j^t=(\overline{CF}_j^t,\overline{T}_j^{t,\mathrm{cov}}),j=1,2,3$，**集群中心表示不同集群活跃度的平均水平，**每个 VU **活跃度**与集群中心的欧几里得距离被计算为

$$e_{u,j}^t=\sqrt{(\overline{CF}_u^t-\overline{CF}_j^t)^2+(\overline{T}_u^{t,\mathrm{cov}}-\overline{T}_j^{t,\mathrm{cov}})^2} \tag{22}$$

每个 VU 将被分配给欧几里得距离最近的集群中，然后通过对该集群中所有 VUs 的活跃度进行平均以更新中心，即

$$v_j^t=(\frac{1}{|V_j^t|}\sum_{u\in V_j^t}\overline{CF}_u^t,\frac{1}{|V_j^t|}\sum_{u\in V_j^t}\overline{T}_u^{t,\mathrm{cov}}) \tag{23}$$

式中：V_j^t 表示分配到集群 j 的 VU 集合，$|V_j^t|$ 表示集群 j 中的 VU 数量。**当集群中心 v_j^t 不再变化，则表明每个 VU 都被分配到了最适合的集群，聚类迭代则会终止。**

（2）内容流行度预测。每个 VU 的历史请求信息被 AE 抽取出来以构造一个内容流行度评价矩阵 H_c，其中 H_c 的第一维度是 VU 的身份特征信息，第二维度是 VU 对内容的评价等级。每个集群 VU 的内容按由高到低进行排序，选择具有最高评价等级的内容 C_j^t 作为预测流行内容。

（3）内容差异化选取。每个集群需要将其预测的流行内容上传到 RSU，为了保证服务的公平性，RSU 需要对不同集群预测的流行内容进行差异化选取。具体来说，在选取流行内容时不仅需要考虑活跃度高的 VU，其他 VU 的请求也能在一定程度上被及时满足。因此集群 j 上传的预测流行内容的权重被计算为

$$\varphi_j^t=\frac{|v_j^t|}{\sum_j|v_j^t|} \tag{24}$$

式中：$\varphi=(\varphi_1^t,\varphi_2^t,\varphi_3^t)$，且 $\varphi_1^t+\varphi_2^t+\varphi_3^t=1$。由于每个 RSU 的缓存容量有限，必须根据权重进行预测内容的选取，以优化存储资源利用率。RSU m 生成的预测内容列表可以被表示为 $C_m^t=\varphi\sum_j C_j^t$。

3.决策算法设计

本文目标在于设计一种高效的内容缓存策略，以实现每个 RSU 内容传输成本的最小化与系统收益的最大化。为此，将缓存替换过程建模为 MDP，并通过量化关键性能指标，引入多智能体 DRL 方法，以联合求解最优缓存策略。

3.1 马尔可夫决策过程

RSU 应该合理替换缓存的内容以满足 VU 最新的内容请求，提供更好的体验。本文将 RSU 的缓存替换表示为 MDP，它由三元组 $(\mathrm{S},\mathrm{A},\mathrm{R})$ 来表示[31]。其中 S 表示环境状态，表示每个 RSU 的变化和 VUs 的状态。A 是所有 RSU 执行决策的集合，确定哪些内容被替换以及如何满足内容请求。R 是 RSU 奖励的集合，定义为在当前状态下执行动作的净收益。对于 RSU m，认为它的三元组 (s_m^t,a_m^t,R_m^t) 由在时隙 t 处的状态、动作和奖励组成。

（1）状态空间 S：由于 RSU m 是基于对自身缓存内容和服务区域内 VUs 的观察来做出替换决策，因此定义状态为 $s_m^t=\{\boldsymbol{p}_m^t,\boldsymbol{r}_m^t\}$，其中 $\boldsymbol{p}_m^t$ 表示每个时隙 t 的本地缓存状态，$\boldsymbol{r}_m^t$ 表示接收到的内容请求状态。

- 将 RSU m 的本地缓存状态表示为 $\boldsymbol{p}_m^t=\{p_{m,1}^t,\cdots,p_{m,c}^t\cdots,p_{m,C}^t\}$，并且 $p_{m,c}^t=1$ 表示内容 c 已经被存储在 RSU m 处，否则 $p_{m,c}^t=0$。
- 把 RSU m 接收到的内容请求状态定义为 $\boldsymbol{r}_m^t=\{r_{m,1}^t,\cdots,r_{m,c}^t\cdots,r_{m,C}^t\}$，其中 $r_{m,c}^t=\sum_{u\in U_m^t}\rho_{u,c}^t$ 表示 RSU m 覆盖范围内容 c 被请求的总数。

（2）动作 A：RSU m 需要确定要缓存或替换的内容，以及如何满足内容请求，因此定义每个 RSU m 中的动作为 $a_m^t=\{\boldsymbol{c}_m^t,\boldsymbol{g}_m^t\}$，其中 $\boldsymbol{c}_m^t$ 为 RSU m 的缓存动作，$\boldsymbol{g}_m^t$ 则表示其缓存服务方式。本文假设 RSU m 在每个时隙 t 内从集合 C 中选择内容进行缓存替换。

（3）奖励 R：在这个问题中，奖励函数与系统收益和平均成本的差直接相关。奖励函数的数学表达式为

$$R_m^t=E_m^t-K^t \tag{25}$$

3.2 基于内容预测的多智能体缓存策略

DRL 在解决协同缓存场景中各类复杂决策问题时已得到广泛应用，其中深度神经网络（Deep neural network, DNN）被用于训练学习模型，并通过与环境的持续交互实现策略优化。然而，随着 VU 数量、RSU 数量及内容数量的增加，缓存决策的维度会随之增长，显著提升了集中式决策的复杂性。为应对这一高维决策难题，本文引入多智能体 DRL 方法。如图 4 所示，在每个 RSU 内部署 DDPG 网络[32]，使各个独立的智能体能够自主与环境进行交互，实现分布式决策。

DDPG 由一个用于制定策略的 Actor 网络和一个用于价值估计的 Critic 网络组成。其借助确定性策略梯度直接优化 Actor 网络，可在连续动作空间中稳定习得确定性最优策略，能平衡探索与利用，引导智能体趋向长期收益更优的决策，避免因随机探索过度或不足陷入局部最优。

在协作的云边缓存车辆网络中，RSU 的决策均用离散值{0,1}来映射。然而，DDPG 算法通常适用于连续动作空间，不能直接输出缓存替换动作和请求处理动作这类离散动作。**为了克服上述限制，本文借鉴文献[33]提出的概率化 DDPG 策略的思路，将 Actor 网络的输出从单一的连续动作扩展为一组连续概率分布向量。具体而言，Actor 网络在状态输入下生成各个离散动作的概率分布 $\bar{a}=\bar{\pi}\left(s;\bar{\theta}\right)$，其中第 i 个元素表示选择第 i 个离散动作的概率；随后通过采样函数 $\psi\left(\bar{a}\right)$ 从该分布中抽取实际执行的离散动作。这种方式保证了策略梯度在离散动作情形下仍然存在，且 DDPG 通过概率化建模的方式实现了对离散动作空间的优化。**

在此基础上，将该离散化方法引入本文 MADDPG 框架，每个智能体各动作的选择实质上是基于 Actor 网络输出的概率分布推导而来。本文将这种改进称为适配离散场景的 MADDPG，它能够在离散动作空间下实现多智能体的高效协同缓存与请求处理。

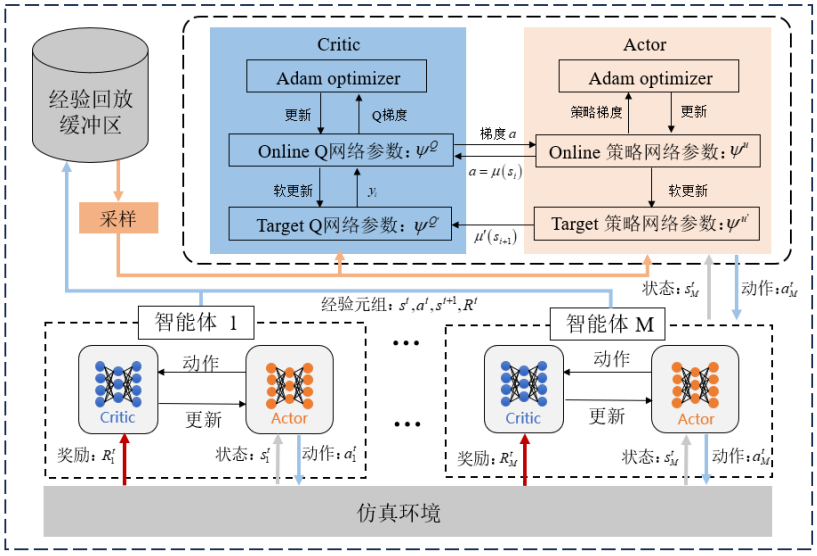

图 4 基于 DDPG 的 MADRL 改进协同缓存算法

Fig.4 Improved collaborative caching algorithm of MADRL based on DDPG

如图 4 所示，每个 RSU 作为一个独立的智能体，均需要部署模型，配置自己的 Actor 和 Critic 网络。此外，还会有目标网络用于稳定训练过程。当前 Actor 网络参数用 ψ^u 表示，目标 Actor 参数网络用 $\psi^{u'}$ 表示，同理 Critic 网络参数用 ψ^Q 表示，目标 Critic 网络参数用 $\psi^{Q'}$ 表示，网络从输入状态空间中的初始状态 s_m^t 生成动作 a_m^t。然后，智能体观察下一个状态 s_m^{t+1} 和这个动作产生的奖励 R_m^t。这四个参数存储在一个经验回放缓冲区中，该缓冲区用于通过随机元组抽样更新网络参数。从经验回放缓冲区中随机抽取元组 $a_m^t,s_m^t,s_m^{t+1},R_m^t$，并利用其来更新 Critic 和 Actor 网络。目标 Q 值计算公式为

$$y_m^t=R_m^t+\vartheta Q'\left(s_m^{t+1},\mu'\left(s_m^{t+1};\psi^{u'}\right);\psi^{Q'}\right) \tag{26}$$

式中：Q' 是目标 Critic 网络，u' 是目标 Actor 网络，ϑ 是折扣因子，s_m^{t+1}、$\mu'\left(s_m^{t+1}\right)$ 是下一状态和下一动作。

Critic 网络的优化目标是最小化以下损失函数

$$L_Q^t=\mathrm{E}\left[\left(Q(s_m^t,a_m^t;\psi^Q)-y_m^t\right)^2\right] \tag{27}$$

此外，Actor 网络通过最大化 Critic 网络的 Q 值来优化策略，其目标函数为

$$J\left(\psi^{a}\right)=\mathrm{E}\left[Q\left(s_{m}^{t},\mu\left(s_{m}^{t};\psi^{u}\right);\psi^{Q}\right)\right] \tag{28}$$

使用梯度上升法更新 Actor 网络

$$\nabla_{\psi^{u}}J\approx\mathrm{E}\left[\nabla_{a_{m}^{t}}Q\left(s_{m}^{t},a_{m}^{t};\psi^{Q}\right)\right]\left|s=s_{m}^{t},a=\mu\left(s_{m}^{t}\right)\nabla_{\psi^{Q}}\mu\left(s_{m}^{t};\psi^{u}\right)\right. \tag{29}$$

目标 Actor 网络$\psi^{u'}$和目标 Critic 网络$\psi^{Q'}$更新过程为

$$\begin{aligned}\psi^{u'}&\leftarrow\vartheta_{1}'\psi^{u}+(1-\vartheta_{1}')\psi^{u'}\\ \psi^{Q'}&\leftarrow\vartheta_{2}'\psi^{Q}+(1-\vartheta_{2}')\psi^{Q'}\end{aligned} \tag{30}$$

式中：ϑ_{1}'和ϑ_{2}'是软更新的步长，用于控制目标网络的更新速率。通过软更新，目标网络的参数变化较为平缓，避免了目标 Q 值的剧烈波动，从而稳定训练过程。

本文提出的具体训练流程如算法 1 所示。

算法 1 基于内容预测的缓存优化

（1）初始化每个 RSU 的经验回放缓冲区 B 、每个 RSU 的策略网络、价值网络和目标 Actor 网络；

（2）for episode=1

（3） for t=1

（4） for RSU m =1

（5） RSU m 观察坏境得到状态s_{m}^{t}，并根据s_{m}^{t}选择动作a_{m}^{t}

（6） RSU 获得奖励R_{m}^{t}，环境进入下一个状态s_{m}^{t+1}

（7） end for

（8） 如果经验回放缓冲区有空间，则在回放缓冲区中存储经验$\left(a_{m}^{t},s_{m}^{t},s_{m}^{t+1},R_{m}^{t}\right)$

（9） 否则，如果经验回放缓冲区已满，则用$\left(a_{m}^{t},s_{m}^{t},s_{m}^{t+1},R_{m}^{t}\right)$取代经验回放缓冲区之前的经验

（10） end for

（11） for RSU m =1

（12） 从经验回放缓冲区 B 中抽取几个经验组

（13） 根据式(26)计算目标 Q 值

（14） 根据式(27)更新 Critic 网络

（15） 根据式(29)用 Critic 网络的梯度来调整 Actor 网络的参数$\nabla_{\psi^{u}}J$

（16） 根据式(30)软更新目标网络$\psi^{u'}$和$\psi^{Q'}$

（17） end for

（18）end for

CRP-BA 预测算法和基于 MADRL 的缓存算法在一个紧耦合反馈回路中运行。首先，CRP-BA 算法聚合经过联邦训练的 AE 提取的潜在特征（如区域请求模式、内容元数据），生成时空流行预测。这些预测作为先验知识注入到 MADRL 智能体的状态空间中，以指导缓存决策。MADDPG 通过将 RSU 视为一个分散的智能体来自适应分配缓存内容，该智能体通过 Q 值和动作策略更新来协调异构的 VU 请求，并将反馈回

CRP-BA 的全局模型。这种反馈触发 VU 组的重新聚类和 CRP-BA 算法中模型聚集权重的重新校准，确保预测缓存周期逐步适应不断变化的网络条件，同时保持长期的公平性。

3.3 算法复杂度分析

本节将分析本文提出的缓存算法的计算复杂度相关问题。在预测阶段，算法复杂度主要体现在通信复杂度方面。在联邦学习框架下，MBS、RSU 与 VU 之间需要频繁交互以实现全局预测模型的收敛。本文系统中共有 K 个 VU、M 个 RSU，以及一个 MBS，局部预测模型的维度为 MK 。因此，其通信复杂度可表示为 $\mathrm{O}(MK)$。在缓存决策阶段，复杂度受状态空间和动作空间维度的影响，关键因素包括内容数量 F 、VU 的数量 K 、以及 RSU 的数量 M 。且缓存决策阶段的计算资源和时间消耗巨大，因此在整体复杂度分析中，决策阶段的开销占据主要比重。

在决策过程中，每个智能体的 Actor 网络的输入包括一组状态，其中包括其本地缓存状态和内容请求状态，因此输入规模为 $2MF$ 。输出包括缓存动作和缓存交付方式，其维度分别为 $M(F+1)$ 和 KMF ，因此总输出规模为 $M(F+1)+KMF$ 。每个智能体使用一个 Critic 网络来估计 Q 函数值。Critic 网络的输入包括全局状态和动作，其维度分别为 $2MF$ 和 $M(F+1)+KMF$ ，并输出一个维度为 1 的 Q 值。根据文献[34]中的复杂性分析方法，假设使用具有固定层数和神经元数量的全连接神经网络，反向传播过程的计算复杂度与输入和输出规模的乘积成正比。对于 Actor 网络，反向传播复杂度表示为 $\mathrm{O}(M^2F^2K)$；对于 Critic 网络而言，其计算复杂度表示为 $\mathrm{O}(M^2F+KMF)$。因此，算法的总计算复杂度可表示为 $\mathrm{O}(M^2F^2K)$。

4.仿真结果与分析

采用 Python 对提出的算法进行仿真，并在城市道路上模拟了一个包括 48 个 VU 和 4 个 RSU 的车辆网络场景作为仿真场景，每个 RSU 的覆盖范围是半径 1 km 的圆。本文的实验是基于 GTX3090 GPU 进行，并使用了 Python 3.8 和 PyTorch 1.12。为了实验更具有实用性，使用了真实数据集 MovieLens 1M[35]。

该数据集包含 6,040 位匿名用户对 3,883 部电影的 1,000,209 个评分，评分范围从 0 到 1，被广泛应用于车联网中的内容缓存实验[36]。每个 VU 随机分配数据集，99.8%用于训练，0.2% 用于测试。除非另有说明，实验的模拟参数详见表 1。

表 1 仿真参数设置

Table 1 Simulation parameter settings

参数	数值
系统带宽 W	100 kHz
RSU m 的存储容量 S_m	200 Mbit
内容 $c \in C$ 的大小 s_c	[5,10] Mbit
RSU m 和 RSU n 之间的有线链路传输速率 $r_{m,n}^t$	15 Mbps
RSU m 的发射功率 P_m	30 dBm
噪声功率 σ^2	-114 dBm

无线链路传输内容的单位成本 α_1	1
有线链路传输内容的单位成本 α_2	0.5
城市交通规定的最大速度 v_h	60 km/h
城市交通规定的最小速度 v_l	50 km/h
经验回放缓冲区大小 B	10000
折扣因子 ϑ	0.90

4.1 预测算法性能分析

为了评估 CRP-BA 在预测准确度方面的表现，将其与三种对比预测算法进行比较：高度活动算法（High activity algorithm, HAA）[37]，通用基本算法（Common basic algorithm, CBA）[10]和无联邦学习的 CRP-BA 算法[38]。需要额外说明的是，为了保证实验的客观性，预测结果均输入到本文提出的 MADDPG 算法中进行决策。

1）HAA：一种优先考虑系统中活跃度最高的 VU 的历史请求的内容预测方法。通过分析 VU 的频繁请求来评估内容的受欢迎程度，并据此对预测的受欢迎程度进行排序，以确定需要缓存的内容；

2）CBA：一种非个性化的内容预测算法，忽略了 VU 行为的动态变化以及长期历史数据，仅依据最近内容请求的频率来进行预测；

3）无联邦学习的 CRP-BA 算法：在基于本文算法的基础上，取消了联邦学习的聚合，每个 VU 均在本地训练全局模型，返回权重列表用以生成每个 VU 个性化的推荐列表，进而生成 RSU 推荐内容。

图 5 比较了所提出的 CRP-BA 在预测流行内容方面的准确性与三种预测算法的对比。所提出的 CRP-BA 一直保持着约 94%的准确率，优于无联邦学习的 CRP-BA 算法，因为联邦学习能增加预测模型的泛化性；其次优于 HAA 和 CBA，因为 HAA 仅聚焦活跃度最高的 VU 历史请求，对其他 VU 需求关注不足，难以全面反映内容真实受欢迎程度，而 CBA 忽略用户行为动态变化与长期历史数据，仅依赖近期请求频率预测，易受短期波动影响而产生偏差。但 CRP-BA 能够在考虑 VU 活跃度的差异的同时提高模型泛化性，从而能够更有效地分析热门内容与 VU 请求内容之间的相关性，进而提高预测的准确性。

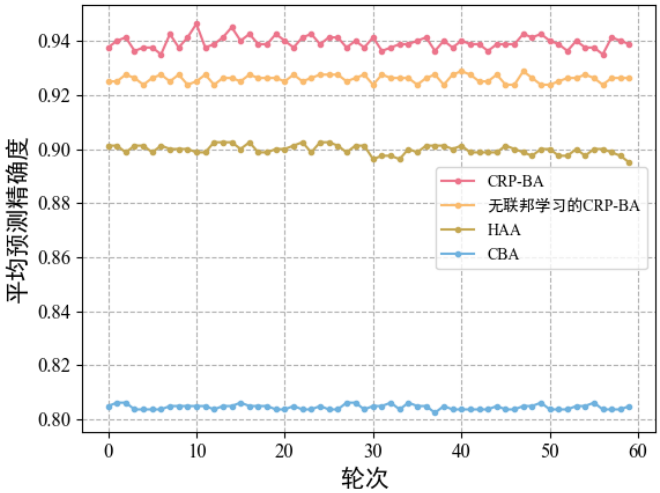

图 5 不同预测算法的预测精确度对比

Fig.5 Comparison of prediction accuracy of different prediction algorithms

4.2 缓存算法消融实验

为了评估各组成部分在所提方案中的实际贡献，本文依次去除相应组件并开展消融实验，以分析其对整体性能的影响。具体而言，本文设置了以下两种消融实验情景。

不聚类的 MADDPG 算法：在基于本文算法的基础上，取消了聚类的设计，每个 VU 不会根据活跃度进行聚类，预测内容按照活跃度排序生成 RSU 推荐内容。

无联邦的 MADDPG 算法：在基于预测对比算法—无联邦学习的 CRP-BA 算法的基础上，将预测内容缓存进 RSU 中，再进行缓存决策。

如图 6 所示，系统的平均收益随着 RSU 缓存容量的增长而逐步增加。尽管三种方法的性能均有所提升，但集成了聚类与联邦学习两部分的 MADDPG 算法始终保持最高收益，并呈现稳步增长态势。相比之下，缺少联邦学习的 MADDPG 算法表现明显较低，其性能不仅落后于同时结合聚类与联邦学习的设计，也低于不采用聚类的方案。在不同缓存容量场景下，本文提出的 MADDPG 算法相较于不聚类的 MADDPG 算法的系统平均收益最高提升了 14%，相较于无联邦的 MADDPG 算法的系统平均收益最高提升 32%。同样的，如图 7 所示，随着 RSU 数量的增长，三种方法的平均系统收益也不断提高。且本文提出的 MADDPG 算法相较于不聚类的 MADDPG 算法的系统平均收益最高提升了 7%，相较于无联邦的 MADDPG 算法的系统平均收益最高提升 15%。

这是因为在缺乏联邦学习的情况下，训练过程相当于各 VU 独立进行，难以学习到全局的偏好模式。而不聚类的算法虽然在结构上更为简化，但仍然保留了联邦学习机制，能够聚合多个 VU 的信息，通常带来更优的流行度预测或内容偏好建模。需要指出的是，尽管不聚类的设计可能会使部分低活跃度车辆在缓存分配上处于劣势，但由于其能够提前学习并利用全体 VU 的偏好信息，因此从整体系统角度来看，其性能仍然优于完全缺少联邦学习的方案。

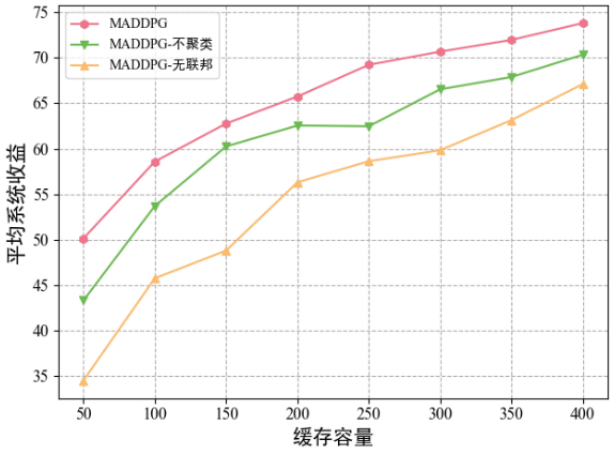

图 6 不同缓存容量下的消融实验对比

Fig.6 Comparison of ablation experiments under different cache capacities

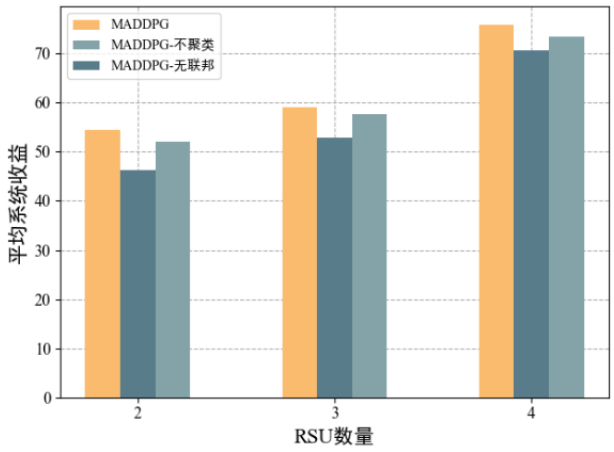

图 7 不同 RSU 数量下的消融实验对比

Fig.7 Comparison of ablation experiments under different numbers of RSUs

4.3 缓存算法性能分析

本文将 MADDPG 与三种决策算法进行比较：基于规则的随机算法（Random）和汤普森采样 (Thompson sampling, TS)[39]，以及基于 DRL 的多智能体深度双 Q 学习网络 (Multi-agent deep double Q-learning network, MADDQN)[14]。此外，将缓存命中率作为一个重要的评估指标，它被定义为本地 RSU 的缓存命中总数和 VU 的请求总数之比，即 $h_i^t = \sum_{u=1}^{U_m^t}\sum_{c=1}^{C}\rho_{u,c}^t g_{u,c}^{m,t} \Big/ U_m^t$ ，其中 U_m^t 表示 RSU m 服务的 VU 数量。

1）Random：随机选择流行内容进行缓存替换，并不考虑内容的受欢迎程度或不同 VU 之间的公平性，完全基于均匀概率分布，而不依赖于任何历史请求或内容特征。

2）TS：基于贝叶斯推断的随机决策算法，会动态估计内容流行度的概率分布，缓存满时通过抽样确定保留和删除的内容，在探索可能受欢迎的新内容与保留已知热门内容间找到平衡。

3）MADDQN：一种在小规模场景中有效的经典 DRL 方法，使用经验回放缓冲区，并定期更新目标网络以稳定训练过程。通过将动作选择与价值估计分离，减少了 Q 值的高估现象。在高维状态空间中，MADDQN 效果显著，但需要大量的离线训练。

（1）收敛性能分析：如图 8 所示，本文将 CRP-BA 预测算法的结果分别通过 DRL 方法 MADDPG 和 MADDQN 进行决策，对比了这两种缓存算法在相同环境下的收敛性能，使用 Reward 作为性能标准，评估了 500 多个回合。图中的收敛曲线描绘了奖励的平均值(实线)和标准差(阴影区域)。

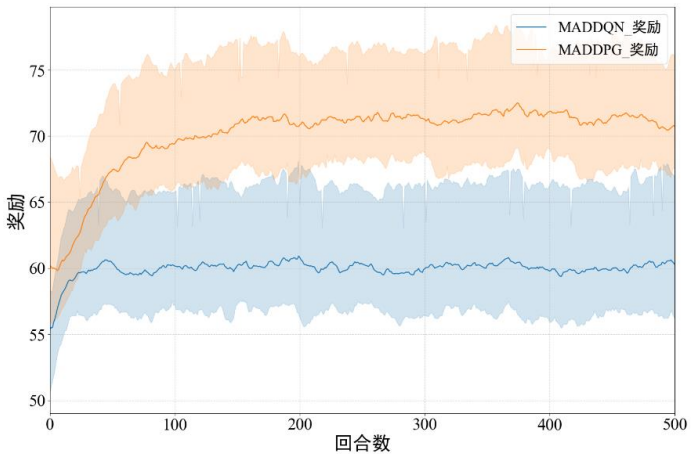

图 8 基于 MADRL 的缓存决策算法的收敛性能

Fig.8 Convergence performance of caching decision algorithm based on MADRL

提出的 MADDPG 算法表现出更稳定的收敛性，奖励值在前 150 回合数中迅速提高并稳定在 82 附近。MADDPG 的探索和学习上表现更优，奖励值高于 MADDQN。相比之下，MADDQN 在复杂环境中难以充分适应，收敛性也弱于 MADDPG。

（2）不同 RSU 缓存容量的影响：

图 9 和图 10 比较了本文所提出的 MADDPG 算法与三种对比算法在不同 RSU 缓存容量下的平均系统收益和缓存命中率方面的表现，其中系统收益指系统净收益。如图所示，随着缓存容量的增大，系统收益及缓存命中率逐渐提高，这是因为单个 RSU 缓存容量增加后，VU 能从本地 RSU 获取更多请求内容，大大增加了 VU 请求内容的命中率，大幅降低了内容交付延迟与成本。此外，MADDPG 始终优于其他三种对比算法，原因在于 MADDQN 在离散场景中易因多智能体 Q 值估计相互干扰，难以稳定学习协同策略，而 MADDPG 能通过持续学习历史交互数据，动态捕捉内容请求规律 VU 行为特征，相比完全随机选择的 Random 算法和仅依赖概率抽样的 TS 算法，更能适应复杂动态的环境，在系统收益上表现更优。

图 11 比较了本文所提出的 MADDPG 算法与三种对比算法在不同 RSU 缓存容量下的平均公平性。可以看出，公平性随着缓存容量的增加而上升；DRL 方法 MADDPG 和 MADDQN 的公平性始终高于 Random 算法和 TS 算法。在基于规则的算法中，Random 完全依赖概率选择，难以保障不同区域 VU 的请求公平性，而 TS 虽有概率调整机制，但缺乏对全局 VU 需求差异的针对性考量，公平性波动较大。相比之下，本文提出的 MADDPG 算法通过联合策略学习，在多智能体协同环境下能够更稳定地兼顾各个 VU 的请求差异，从而实现更优的公平性表现。值得一提的是，MADDQN 在公平性上虽略低于 MADDPG，但依旧显著优于传统策略，体现了其在多智能体环境中较强的适应性与调度能力。

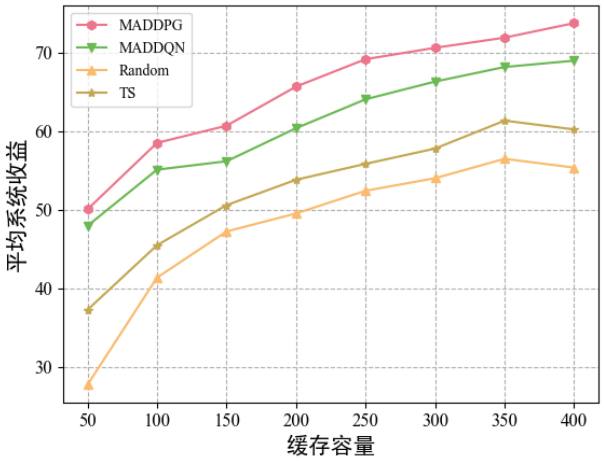

图 9 不同缓存容量下的平均系统收益对比

Fig.9 Comparison of average system revenue under different cache capacities

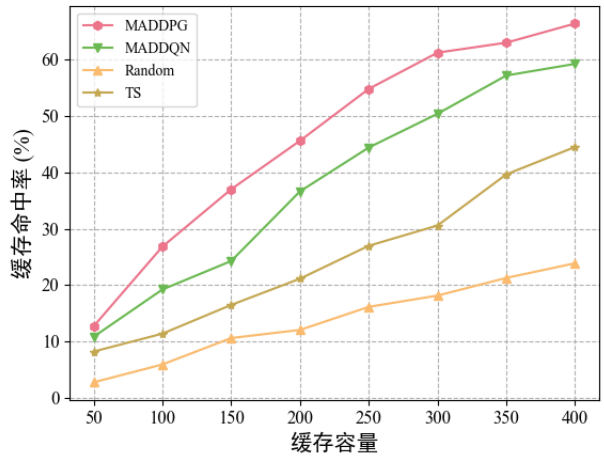

图 10 不同缓存容量下缓存命中率对比

Fig.10 Comparison of cache hit rates under different cache capacities

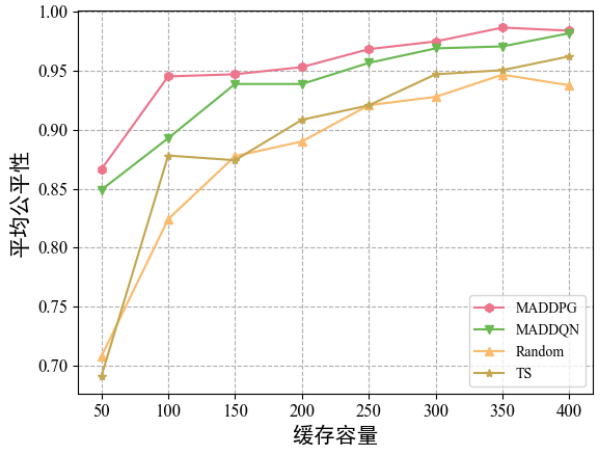

图 11 不同缓存容量下的平均公平性对比

Fig.11 Comparison of average fairness under different cache capacities

（3）不同 RSU 数量的影响：

如图 12 所示，随着 RSU 的数量从 2 增加到 4，基于 DRL 的两种缓存决策算法在平均系统收益方面始

终优于两种基于规则的缓存决策算法，这归因于基于规则的算法依赖固定逻辑和简单概率模型，无法随 RSU 数量增加动态调整策略，不能应对多 RSU 间的协同缓存需求，而 DRL 算法的环境感知和持续学习能力，能制定更贴合全局收益的决策。而另一方面， MADDPG 表现出最高的平均系统收益，比 MADDQN 高出了 8%。这主要因为 RSU 数量增加会导致决策空间的快速扩展，MADDQN 在离散动作空间中对高维度决策的适配性较弱，易出现 Q 值估计偏差，而提出的 MADDPG 可以更出色的捕捉和适应复杂环境特征。

此外，图 13 中比较了不同算法在不同的 RSU 数量下缓存命中率方面的性能。本文提出的 MADDPG 可以达到 65%以上的缓存命中率，而 MADDQN 只达 60%左右。因为 MADDPG 的 Critic 网络能利用全局状态信息更精准地评估动作价值，引导 Actor 网络生成更优的缓存替换策略，减少因局部决策偏差导致的缓存资源浪费；但 MADDQN 在多 RSU 场景下的 Q 值更新易受其他智能体动作干扰，难以稳定学习最优策略。

图 14 展示了不同 RSU 数量下各算法的平均公平性表现。可以观察到，当 RSU 数量从 2 增加至 4 时，各算法的公平性指标波动幅度较小，整体保持相对稳定。其中，MADDPG 始终处于领先地位，MADDQN 次之，Random 与 TS 的公平性表现依次落后。MADDPG 依托 Actor-Critic 架构，可整合多 RSU 的全局状态，通过全局价值评估引导策略学习，即便 RSU 数量增加，仍能保障不同车载用户 VU 的服务公平性；而 MADDQN 依赖各 RSU 的局部状态更新 Q 值，缺乏全局协同易导致局部最优挤占公平性，可能会忽视部分区域 VU 的请求；Random 的纯随机决策与 TS 的局部概率优化则因无全局感知能力，难以稳定保障公平性，故表现最差。

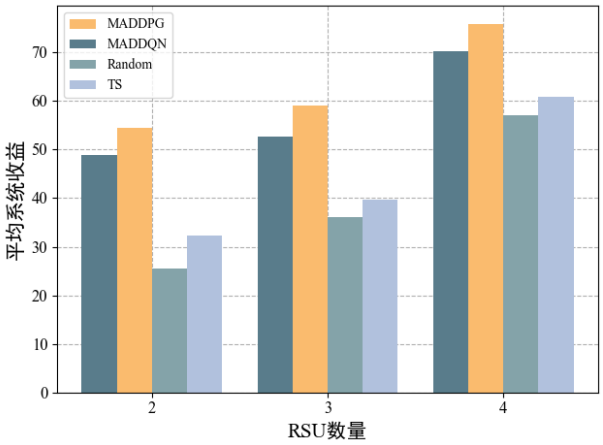

图 12 不同 RSU 数量下的平均系统收益对比

Fig.12 Comparison of average system revenue under different RSU numbers

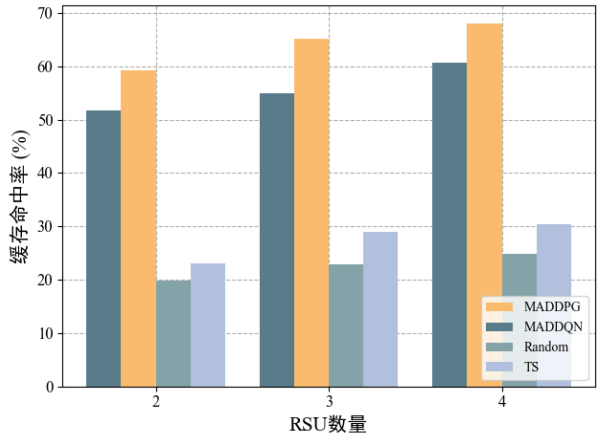

图 13 不同 RSU 数量下的缓存命中率对比

Fig.13 Comparison of cache hit rates under different RSU numbers

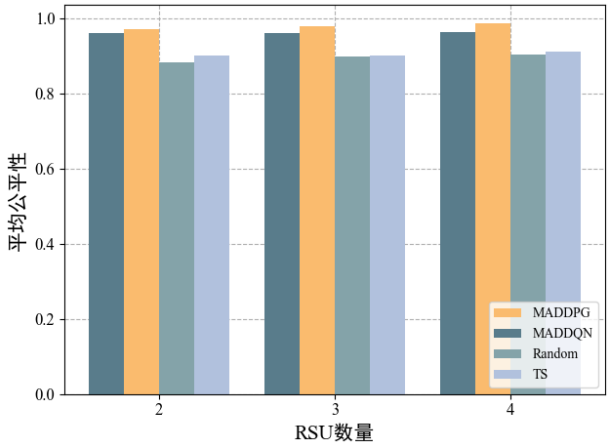

图 14 不同 RSU 数量下的平均公平性对比

Fig.14 Comparison of average fairness under different RSU numbers

（4）不同 VU 数量的影响：

如图 15、图 16、图 17 所示，本文研究了不同数量的 VU 对系统收益和缓存命中率以及系统公平性的影响。图 15 比较了四种缓存决策算法在不同密度下的系统收益，随着 VU 数量从 40 增加到 80，MADDPG、MADDQN、Random、TS 的平均系统收益都呈现出明显的上升趋势。这表明内容请求量的增加有助于提高收益。然而，当 VU 数量超过 80 时，有限的通信带宽开始制约内容的及时交付，导致平均收益下降。在所有场景中，本文提出的 MADDPG 始终优于其他基准算法，这得益于其在离散化处理后仍能够保持对复杂缓存决策的精细建模能力，从而在资源有限的情况下实现更高效的分配与利用。在图 16 中，缓存命中率对 VU 数量的变化相对不敏感。不过，MADDPG 在所有条件下都保持了卓越的性能，这表明它能够准确捕捉 VU 的内容偏好，并确保缓存命中率始终处于较高水平。

如图 17 所示，随着 VU 数量的增加，系统的平均公平性逐渐下降。这是因为 VU 数量增加导致内容请求量显著增长，而有限的系统资源加剧了分配不均的问题。此外，VU 数量的增加带来了系统内容总请求量的提升，即使缓存命中率与 VU 数量较少场景下的缓存命中率相差不大，系统总收益仍表现为上升趋势。但与此同时，系统公平性下降带来的公平性成本以及通信带宽受限导致的时延惩罚，均会对收益增长形成制约。最终，系统收益随着 VU 数量如图 15 所示呈现先升后降的趋势。

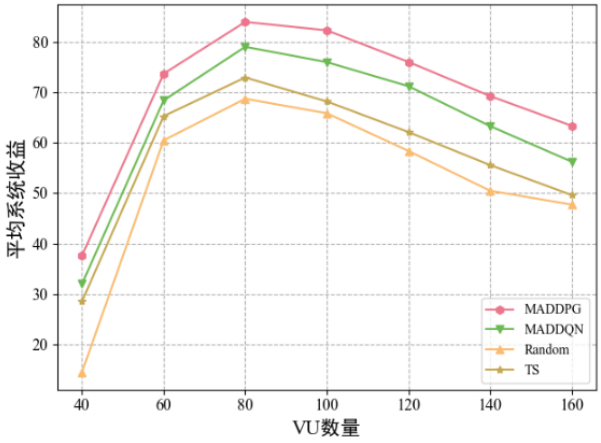

图 15 不同 VU 数量下的平均系统收益对比

Fig.15 Comparison of average system revenue under different VU numbers

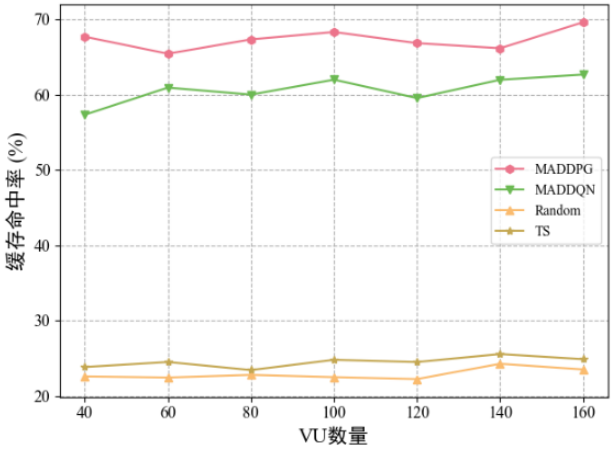

图 16 不同 VU 数量下的缓存命中率对比

Fig.16 Comparison of cache hit rates under different VU numbers

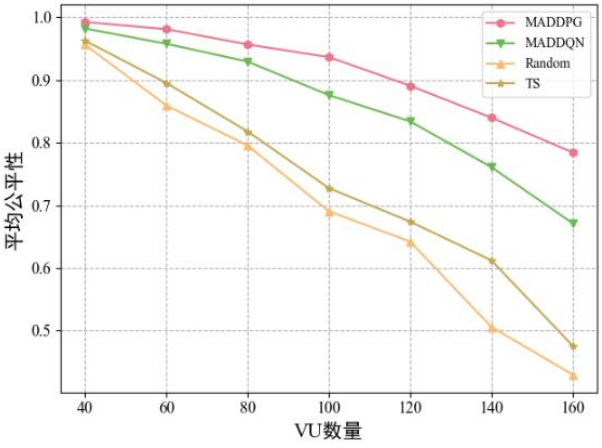

图 17 不同 VU 数量下的平均公平性对比

Fig.17 Comparison of average fairness under different VU numbers

5.结论

本文研究了一种结合了 MADRL 和联邦学习的协同缓存策略，特别注重在确保服务公平性的同时实现长期系统收益的最大化。为此，我们提出了 CRP-BA 算法以进行准确的热门内容预测，并借助三层联邦学

习框架实现高效的模型更新。CRP-BA 算法利用历史请求数据提取潜在特征，并生成热门内容的排序列表。此外，我们设计了一种基于内容预测的 MADDPG 算法来解决所提出的优化问题，从而实现高效的内容缓存和交付决策。大量的模拟结果验证了所提出方法的有效性和优越性。未来的工作将探索交通流量预测、基于大型模型的内容推荐以及为动态车辆网络开发时间敏感或与移动性相关的公平性指标。

参考文献：

[1] He P, Li S, Wang C, et al. Social-aware collaborative caching in edge-user networks: A joint multi-graph approach[J]. IEEE Transactions on Network Science and Engineering, 2024, 11(5): 4938-4950.

[2] Zhao Y, Xiao A, Wu S, et al. Adaptive partitioning and placement for two-layer collaborative caching in mobile edge computing networks[J]. IEEE Transactions on Wireless Communications, 2024, 23(8): 8215-8231.

[3] Liu L, Chen C, Pei Q, et al. Vehicular edge computing and networking: A survey[J]. Mobile networks and applications, 2021, 26(3): 1145-1168.

[4] Liu Y, Yang C, Chen X, et al. Joint hybrid caching and replacement scheme for UAV-assisted vehicular edge computing networks[J]. IEEE Transactions on Intelligent Vehicles, 2023, 9(1): 866-878.

[5] Li X, Wang X, Wan P J, et al. Hierarchical edge caching in device-to-device aided mobile networks: Modeling, optimization, and design[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(8): 1768-1785.

[6] Teng Z, Fang J, Liu Y. Combining Lyapunov optimization and deep reinforcement learning for D2D assisted heterogeneous collaborative edge caching[J]. IEEE Transactions on Network and Service Management, 2024, 21(3): 3236-3248.

[7] Ahmed M, Traverso S, Giaccone P, et al. Analyzing the performance of LRU caches under non-stationary traffic patterns[J]. arXiv preprint arXiv:1301.4909, 2013.

[8] Jaleel A, Theobald K B, Steely Jr S C, et al. High performance cache replacement using re-reference interval prediction (RRIP)[J]. ACM SIGARCH computer architecture news, 2010, 38(3): 60-71.

[9] Hoang D T, Niyato D, Nguyen D N, et al. A dynamic edge caching framework for mobile 5G networks[J]. IEEE Wireless Communications, 2018, 25(5): 95-103.

[10] Ndikumana A, Tran N H, Kim K T, et al. Deep learning based caching for self-driving cars in multi-access edge computing[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(5): 2862-2877.

[11] Safavat S, Rawat D B. Improved multiresolution neural network for mobility-aware security and content caching for internet of vehicles[J]. IEEE Internet of Things Journal, 2023, 10(20): 17813-17823.

[12] Yu Z, Hu J, Min G, et al. Mobility-aware proactive edge caching for connected vehicles using federated

learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(8): 5341-5351.

[13] Huang J, Zhang M, Wan J, et al. Joint data caching and computation offloading in UAV-assisted Internet of Vehicles via federated deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2024, 73(11): 17644-17656.

[14] Wu H, Jin J, Ma H, et al. Federation-based deep reinforcement learning cooperative cache in vehicular edge networks[J]. IEEE Internet of Things Journal, 2023, 11(2): 2550-2560.

[15] Tharakan K S, Dahrouj H, Kouzayha N, et al. Personalized federated learning for cellular VR: Online learning and dynamic caching[J]. IEEE Transactions on Communications, 2025.

[16] Wang R, Kan Z, Cui Y, et al. Cooperative caching strategy with content request prediction in Internet of Vehicles[J]. IEEE Internet of Things Journal, 2021, 8(11): 8964-8975.

[17] Sun C, Li X, Wen J, et al. Federated deep reinforcement learning for recommendation-enabled edge caching in mobile edge-cloud computing networks[J]. IEEE Journal on Selected Areas in Communications, 2023, 41(3): 690-705.

[18] 张健,刘鹏博,汤健.联合 WPT 和 MEC 的无线传感网时延优化算法[J].数据采集与处理,2025,40(01):163-175.

ZHANG Jian, LIU Pengbo, TANG Jian. Delay Optimization Algorithm in Wireless Sensor Networks Combining WPT and MEC [J]. Journal of Data Acquisition and Processing, 2025,40(01):163-175.

[19] Singh P, Hazarika B, Singh K, et al. DRL-based federated learning for efficient vehicular caching management[J]. IEEE Internet of Things Journal, 2024, 11(21): 34156-34171.

[20] Zhou Y, Wang F, Shi Z, et al. An efficient deep reinforcement learning-based automatic cache replacement policy in cloud block storage systems[J]. IEEE Transactions on Computers, 2023, 73(1): 164-177.

[21] Wu Q, Zhao Y, Fan Q, et al. Mobility-aware cooperative caching in vehicular edge computing based on asynchronous federated and deep reinforcement learning[J]. IEEE Journal of Selected Topics in Signal Processing, 2022, 17(1): 66-81.

[22] Tian W, Chen Y, Ke F, et al. Deep Reinforcement Learning-Based Mobile-Aware Edge Cooperative Caching Scheme in the Internet of Vehicles[J]. IEEE Transactions on Vehicular Technology, 2025, 74(3), 5053-5068.

[23] Wu Q, Wang W, Fan P, et al. Cooperative edge caching based on elastic federated and multi-agent deep reinforcement learning in next-generation networks[J]. IEEE Transactions on Network and Service Management, 2024, 21(4): 4179-4196.

[24] Sun C, Li X, Wang C, et al. Hierarchical deep reinforcement learning for joint service caching and

computation offloading in mobile edge-cloud computing[J]. IEEE Transactions on Services Computing, 2024, 17(4): 1548-1564.

[25] Yang Y, Lou K, Wang E, et al. Multi-agent reinforcement learning based file caching strategy in mobile edge computing[J]. IEEE/ACM Transactions on Networking, 2023, 31(6): 3159-3174.

[26] Araf S, Saha A S, Kazi S H, et al. UAV assisted cooperative caching on network edge using multi-agent actor-critic reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2022, 72(2): 2322-2337.

[27] Wang Y, Liu Z, Geng C, et al. Distributed Multi-Agent Reinforcement Learning on a Hierarchical Game Model for Railway Engineering Data Collaborative Edge Caching[J]. IEEE Transactions on Intelligent Transportation Systems, 2025, 26(2): 2643-2655.

[28] Goemans M, Li L E, Mirrokni V S, et al. Market sharing games applied to content distribution in ad-hoc networks[C]//Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing. 2004: 55-66.

[29] Wu J, Xu X, Cui G, et al. Fairness-Aware Budgeted Edge Server Placement for Connected Autonomous Vehicles[J]. IEEE Transactions on Mobile Computing, 2025.

[30] Zhou J, Chen F, He Q, et al. Data caching optimization with fairness in mobile edge computing[J]. IEEE Transactions on Services Computing, 2022, 16(3): 1750-1762.

[31] Wang Y, Fang J, Cheng Y, et al. Cooperative end-edge-cloud computing and resource allocation for digital twin enabled 6g industrial iot[J]. IEEE Journal of Selected Topics in Signal Processing, 2023, 18(1): 124-137.

[32] Qiao G, Leng S, Maharjan S, et al. Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks[J]. IEEE Internet of Things Journal, 2019, 7(1): 247-257.

[33] Ding R, Xu Y, Gao F, et al. Trajectory design and access control for air–ground coordinated communications system with multiagent deep reinforcement learning[J]. IEEE Internet of Things Journal, 2021, 9(8): 5785-5798.

[34] Sun C, Wu X, Li X, et al. Cooperative computation offloading for multi-access edge computing in 6G mobile networks via soft actor critic[J]. IEEE Transactions on Network Science and Engineering, 2021, 11(6): 5601-5614.

[35] Harper F M, Konstan J A. The movielens datasets: History and context[J]. Acm transactions on interactive intelligent systems (tiis), 2015, 5(4): 1-19.

[36] Jiang K, Cao Y, Song Y, et al. Asynchronous federated and reinforcement learning for mobility-aware edge

caching in IoV[J]. IEEE Internet of Things Journal, 2024, 11(9): 15334-15347.

[37] Zhou X, Liang W, Kawai A, et al. Adaptive segmentation enhanced asynchronous federated learning for sustainable intelligent transportation systems[J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(7): 6658-6666.

[38] Yuan X, Chen J, Zhang N, et al. A federated bidirectional connection broad learning scheme for secure data sharing in Internet of Vehicles[J]. China Communications, 2021, 18(7): 117-133.

[39] Cui L, Su X, Ming Z, et al. CREAT: Blockchain-assisted compression algorithm of federated learning for content caching in edge computing[J]. IEEE Internet of Things Journal, 2020, 9(16): 14151-14161.

作者简介：

胡前灵（2002-），女，硕士研究生，研究方向：边缘计算、数据缓存，E-mail：1224014011@njupt.edu.cn。

王宇翱（1997-），男，博士研究生，研究方向：云边端智能协同，E-mail：wya@njupt.edu.cn。

孙洪波（1985-），男，实验师，研究方向：边缘智能，E-mail：sunhb@njupt.edu.cn。

郭永安（1981-），**通信作者**，男，教授，研究方向：智能信息处理与通信，E-mail：guo@njupt.edu.cn。