基于竞争双深度Q学习的边缘计算服务部署优化方法

王烨[1]，冉琼慧子[1]，徐悦甦[2]，高洪皓[1]
（1.上海大学计算机工程与科学学院，上海 200444；2.西安电子科技大学计算机科学与技术学院，西安 710126）

摘要：针对车路协同网络所面临的边缘计算服务部署非均衡及资源覆盖受限的问题，提出了一种基于残差图注意力网络与竞争双深度Q学习的边缘计算服务器智能部署方法。首先，通过基于双重邻接矩阵与多维特征编码的V2X环境建模方法将RSU网络转化为结构化图数据。其次，构建了一种融合残差连接与注意力机制的图神经网络提取节点间复杂的空间依赖关系。接着，设计了一种基于竞争双深度Q网络的序列部署决策框架，通过价值函数分解与双网络更新机制来提升Q值估计精度，并结合多目标导向的奖励函数引导策略收敛。实验结果表明，相较于现有方法，所提出的方法改善了5.1%的服务覆盖率、1.4%的负载均衡度和17%的传输延迟。

关键词：车路协同；边缘计算；注意力机制；图神经网络；竞争双深度Q学习
中图分类号：TP309
文献标志码：A
doi：10.11959/j.issn.1000

Optimization method for edge computing service deployment based on dueling double deep Q-learning

Wang Ye[1], Ran Qionghuizi[1], Xu Yueshen[2], Gao Honghao[1]
1. School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
2. School of Computer Science and Technology, Xidian University, Xi'an 710126, China

Abstract: To address the issues of uneven deployment and limited resource coverage of edge computing server for Vehicle-to-Everything (V2X), this paper proposes an intelligent deployment method for edge computing servers based on a residual graph attention network and dueling double deep Q-learning. First, the roadside unit (RSU) network is transformed into structured graph data through a V2X environment modeling method that utilizes a dual adjacency matrix and multi-dimensional feature encoding. Second, a graph neural network integrating residual connection and attention mechanism is constructed to extract complex spatial dependencies among nodes. Third, a sequential decision-making framework based on dueling double deep Q-learning is designed, which improves the accuracy of Q-value estimation through value function decomposition and a dual-network update mechanism, while guiding policy convergence using a multi-objective reward function. Experimental results show that compared to existing methods, the proposed method improves service coverage by 5.1% and load balancing by 1.4%, and reduces transmission latency by 17%.

Key words: Vehicle-to-Everything, edge computing, attention mechanism, graph neural network, dueling double deep Q-learning

0 引言

随着智能交通系统的快速发展，车路协同网络（vehicle-to-everything, V2X）已成为实现车辆协同感知、决策与控制的关键支撑技术[1-3]。V2X通过

收稿日期：XXXX–XX–XX；修回日期：XXXX–XX–XX
通信作者：高洪皓，gaohonghao@shu.edu.cn
基金项目：国家自然科学基金资助项目（No.92367103, No.2022YFF0902500）
Foundation Items：The National Natural Science Foundation of China (No.92367103, No.2022YFF0902500)

车对车（vehicle-to-vehicle, V2V）、车对基础设施（vehicle-to-infrastructure, V2I）、车对行人（vehicle-to-pedestrian, V2P）和车对网络（vehicle-to-network, V2N）等多种通信模式实现信息协作[4-6]。在V2V通信场景中，车辆可以向周围车辆传输实时信息并为驾驶决策提供参考，从而帮助预防交通事故。在V2I通信中，RSU向车辆发送交通信号状态和道路施工信息。V2P通信确保行人和车辆之间的信息交换。V2N将车辆连接到云端，实现广域数据共享和高计算任务的卸载。这种架构使交通系统能够从单车独立决策发展到多因素协作智能，从而实现高效的交通管理和资源分配。然而，V2X中的车辆终端会持续产生大量具有严格时延要求的计算任务，如环境感知、路径规划与碰撞预警等。传统云计算模式因其集中式处理架构与远距离传输特性，难以满足其实时性需求[7-8]。

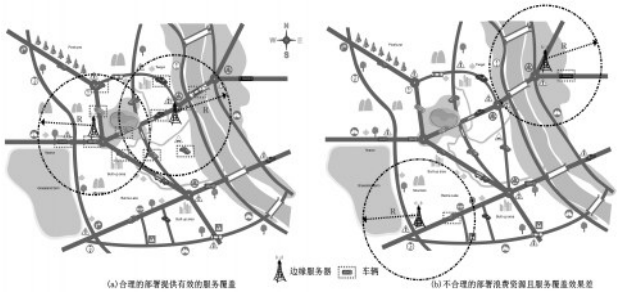


图1　边缘计算服务器部署位置的示例

为此，边缘计算被引入至V2X网络，通过在网络边缘部署计算与存储资源使得数据能够在近源端处理，如路侧单元（roadside unit, RSU），从而有效降低端到端时延[9-11]。边缘计算与V2X网络深度融合所构建的“终端-边缘-云”三级协作架构已成为解决智能交通场景中实时数据处理与低延迟服务响应等核心需求的关键技术方案。在该架构中，边缘计算服务器的部署位置决定了边缘计算服务的覆盖范围、负载分布均衡性以及端到端传输延迟，进而影响V2X网络的服务质量[12-13]。然而，RSU节点的位置通常呈现非均匀分布特征且不同RSU覆盖范围内的车辆数量、业务类型存在差异显著。如图1所示，边缘计算服务器部署位置选择不当，不仅会导致大量高负载区域处于计算服务覆盖盲区，数据被迫回传云端，从而增加延迟；还可能导致多个边缘计算服务器覆盖重叠或集中在低负载区域，造成计算资源浪费和负载失衡。如何在有限的部署资源约束下，协同优化计算服务覆盖率、负载均衡度以及端到端延迟，是构建高效V2X边缘计算系统的关键挑战。

因此，本文聚焦于边缘计算服务器部署非均衡及资源覆盖受限的问题，提出了一种面向V2X场景深度适配的残差图注意力竞争双深度Q学习方法。该方法利用残差图注意力网络提取节点间复杂的空间依赖关系，并采用竞争双深度Q网络进行序列决策优化，实现了计算服务覆盖率最大化、负载分布均衡化与端到端延迟最小化的协同优化目标。本文的贡献总结如下：

（1）提出了一种面向V2X网络的双重邻接矩阵与多维特征编码方法。该方法通过融合通信距离约束与k近邻策略构建双重邻接矩阵以应对V2X无线通信约束与服务连续性需求，并设计了包含静态属性、动态部署状态与综合决策特征的8维特征向量，为后续特征提取和智能决策提供数据基础。

（2）构建了一种融合残差连接与注意力机制的深层图神经网络。该网络利用注意力机制自适应学习RSU节点间的差异化关联权重，并结合残差连接与正则化机制来捕获多跳空间依赖关系的同时保障深层网络的训练稳定性，实现了从局部邻域信息到全局图级特征的高效提取。

（3）设计了一种基于分层奖励引导的竞争双深度Q网络序列部署决策框架。该框架利用价值函数分解与双网络更新机制提升Q值估计的准确性与决策稳定性。在此基础上，结合分层多目标导向的奖励函数引导智能体学习全局最优部署策略，从而实现计算服务覆盖率、负载均衡度与平均延迟的协同优化。

1　相关工作

边缘计算以其轻量级和低延迟的特点，能够满足V2X网络对低延迟服务响应的需求[14-15]。例如，Ning等[16]提出了一种能量敏感的移动边缘计算调度框架，从而有效的降低了能耗与延迟。然而，随着V2X网络中车辆数量和车载应用数的不断增加，边缘计算服务器部署位置的影响将越发显著。不合理的部署边缘计算服务器可能会导致边缘计算服务器负载不均衡，甚至会造成网络性能的下降。为此，Wang等[17]采用混合整数规划，并通过考虑服务端的工作负载平衡以及用户与服务端之间的访问

延迟，找到服务端部署的最优解。作为多目标优化算法的典型代表，遗传算法被广泛应用于部署问题。例如，Zhao等[18]提出了一种改进的多目标非支配排序遗传算法。具体而言，讨论了智慧城市中的k个设备的部署问题，其目标是在约束条件下改善负载均衡和系统延迟。Cao等[19]通过构建涵盖传输延迟、负载平衡、能耗和边缘计算服务器数量等多个指标的部署优化模型来确定边缘计算服务部署。Chang等[20]将边缘计算服务器部署问题拆分成确定边缘计算服务器的位置和每个边缘计算服务器的覆盖范围。此外，利用启发式算法将全局问题分解成规模可控的局部问题并进行求解，从而实现多目标的协同优化。

深度强化学习结合了深度学习的感知能力和强化学习的决策能力，在云-边计算场景中的服务卸载等领域得到了越来越多的关注。Lu等[21]聚焦于边缘计算服务器的负载均衡问题，提出了一种基于深度强化学习的多目标边缘计算服务器部署方案，旨在提高车联网的覆盖率、均衡负载并降低任务传输的平均延迟。实验结果证实了该方案在覆盖率及延迟等方面的显著优势。Meng等[22]提出了一种面向物联网的异构边缘计算服务部署方法，通过一种基于距离和工作负载的边缘服务器集群划分方法来提高部署效率和最小化通信延迟。在此基础上，提出了一种基于层次过程分析的异构边缘计算服务器部署方法来确定最佳部署策略。Zhou等[23]聚焦于优化边缘计算服务器间的负载平衡，提出了一种基于多智能体强化学习的边缘计算服务器部署策略，名为CKM-MAPPO。该策略首先利用k均值算法确定边缘计算服务器数量与初始位置。然后，通过多智能体强化学习算法来确定其最佳位置。为了平衡边缘计算服务部署成本与服务质量，Fan等[24]提出了一种综合考虑多因素的部署优化模型，包括卸载延迟、能耗以及安全性等。在此基础上，提出了一种基于Two-Arch2的生成对抗网络来求解多目标优化问题，名为WGTwo-Arch2。此外，还利用了一种多项式变换策略和一种具有自适应范数的多样性选择策略来平衡算法的探索和开发能力。Zhou等[25]利用结合改进的光谱聚类和Q学习的边缘计算服务器部署方法ISC-QL，来确保网络整体性能的改进。该方法包括聚类阶段和部署阶段。聚类阶段根据基站的地理分布和用户数量进行聚类来确定初始部署位置。在部署阶段，采用Q学习方法优化初始部署位置并确定最佳部署位置。

基于上述分析，现有部署方案多通过整数规划、启发式算法和强化学习方法来进行。然而，传统的整数规划和启发式算法在面临大规模部署时，其计算能力和时间成本往往是难以满足需求的。其次，现有的边缘计算服务器部署方案未结合V2X中RSU的无线通信约束和服务连续性需求进行建模，从而导致拓扑表达能力难以适配真实车路场景。此外，负载均衡、部署数量、覆盖范围以及动

表1　　现有方案对比

类型	文献	方法	目标	考虑因素
传统启发式的方法	[16]	移动边缘计算调度	降低了能耗与延迟	能量平衡、传输延迟
	[17]	混合整数规划	服务端部署最优解	服务端的工作负载平衡、用户与服务端之间的访问延迟
	[18]	多目标非支配排序遗传算法	改善负载均衡和系统延迟	负载平衡、部署数量
	[19]	多目标优化算法	最优边缘计算服务部署	传输延迟、负载平衡、能耗等
	[20]	多目标协同优化算法	边缘计算服务器的位置和覆盖范围	部署成本、时间成本和传输延迟
深度强化学习方法	[21]	深度Q网络	改善覆盖率、均衡负载和平均延迟	覆盖率、传输延迟
	[22]	边缘服务器集群划分方法	确定最佳部署策略	部署成本、传输延迟
	[23]	多智能体强化学习CKM-MAPPO	确定最佳部署数量和部署位置	负载平衡
	[24]	基于Two-Arch2的生成对抗网络	平衡边缘服务部署成本与服务质量	卸载延迟、能耗以及安全性
	[25]	光谱聚类和Q学习	确定最佳部署位置	地理分布、用户数量、负载平衡

态特性的影响很少被全面考虑，且奖励函数无法解决长序列部署决策的稀疏奖励问题，其多目标协同优化能力不足。因此，针对V2X网络中边缘计算服务器部署所面临的资源受限、负载分布不均及拓扑复杂等具体挑战，需要一种适配V2X网络特性的定制化多目标优化部署方案来实现边缘计算服务器部署位置的自适应选择优化。

2　系统模型

2.1　端-边-云计算架构

V2X场景中的端-边-云计算系统是一个典型的三层协作架构，旨在通过将计算资源迁移至网络边缘来解决传统云计算在处理海量、实时车辆数据所面临的高延迟和带宽瓶颈问题。该架构综合考虑V2X场景中RSU地理分布、车辆移动和服务需求动态变化等特性，从而实现边缘计算服务器与RSU、车辆之间的协同调度，最终达成最大化计算服务覆盖率、均衡负载分布和降低端到端延迟的优化目标。

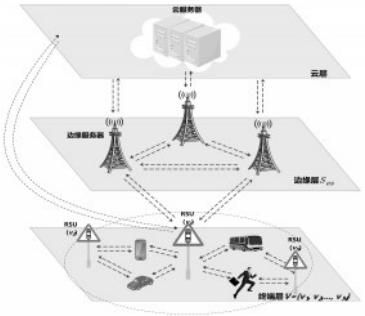


图2　端-边-云计算架构

端-边-云计算架构包括终端层，边缘层和云层，如图2所示。终端层由车辆和RSU组成。车辆作为移动终端产生服务请求，RSU负责收集其覆盖范围内的数据流量，包括车辆状态、交通流量、环境感知信息及潜在的计算任务。对于有N个RSU的终端层，其集合表示为$V=\{v_1,v_2,...,v_N\}$，第i个

RSU在单位时间内会获取数据负载L_i，其大小取决于覆盖区域的交通密度和业务类型。终端层的RSU并不具备强大的数据处理能力，主要承担数据汇聚和上行转发的作用。边缘层由部署在特定RSU节点上的边缘计算服务器组成，其主要功能是接收一个或多个邻近RSU上传的数据，并进行实时处理与分析。考虑到资源受限的情况，需要在有限个RSU上部署边缘计算服务器，其集合表示为S_{es}。云层是由云服务器组成，并拥有丰富的计算和存储资源，但处理延迟显著高于边缘处理。云服务器主要处理两种数据：未被任何边缘计算服务器覆盖的RSU数据和需要全局协调的复杂计算任务。在该架构中，数据优先进行边缘计算处理。具体来说，当RSU与边缘计算服务器的距离小于通信半径$R_{\max}$，则该RSU的数据会被分配给最近的边缘计算服务器进行边缘计算处理。当RSU不在边缘计算服务器的覆盖范围内，则数据会被传输至云端处理。

2.2　覆盖模型

V2X网络中边缘计算服务器的覆盖能力是边缘部署的关键目标。然而，边缘计算资源的约束使得部署策略必须优先覆盖高需求区域。因此，本文构建的覆盖模型主要用于描述边缘计算服务器服务范围及RSU节点的覆盖状态。该模型基于地理距离阈值来判定服务覆盖关系并以数据量为权重计算系统的有效覆盖率，如图3所示。

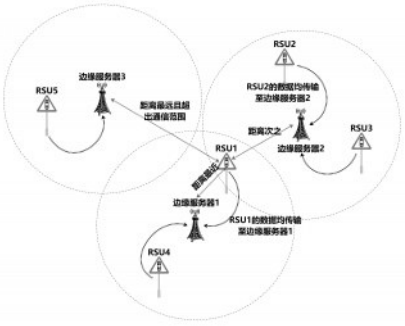


图3　覆盖模型

对于有N个RSU的网络，其RSU的位置坐标集合为$\{(lon_i,lat_i)|i=1,2,...,N\}$，其中$lon_i$和$lat_i$分别是RSU的经度和纬度。已部署的边缘计算服务器集合为$S_{es}\subseteq\{1,2,...,M\}$。对于第$i$个RSU和第$j$个边缘计算服务器之间的地理位置距离需要考虑到地球曲率的影响。因此，采用球面距离公式计算距离如下：

$$d(v_i,v_j)=\sqrt{(\Delta lon_{ij})^2+(\Delta lat_{ij})^2} \tag{1}$$

$$\Delta lon_{ij}=(lon_i-lon_j)\times 111000\times\cos\theta \tag{2}$$

$$\Delta lat_{ij}=(lat_i-lat_j)\times 111000 \tag{3}$$

其中，Δlon_{ij}和Δlat_{ij}分别是经纬度差，常数

111000是地球表面每度的平均距离，θ 是纬度弧度值用来确保距离计算的准确性。若RSU与已部署的边缘计算服务器的最大距离小于通信半径，则该RSU被覆盖，其判定条件表示为：

$$\text{Covered}(v_i,v_j) = \mathbb{I}(d(v_i,v_j) \leqslant R_{\max}) \tag{4}$$

其中，$\mathbb{I}(\cdot)$是指示函数。一个RSU只要被至少一个边缘计算服务器覆盖，即被认为接入了边缘计算服务。因此，定义RSU节点i的被覆盖状态为：

$$c_i = \mathbb{I}(\min_{j \in S_{es}} d(v_i,v_j) \leqslant R_{\max}) \tag{5}$$

其中，c_i 表示当前RSU节点是否已被至少一个已部署的边缘计算服务器覆盖。$c_i = 1$表示RSU节点已经被部署的边缘计算服务器覆盖。在V2X网络中，整体覆盖性能不能简单的以被覆盖RSU数量衡量，而应结合各RSU的数据负载。基于此，服务覆盖率（Coverage Ratio, CR）定义为被覆盖RSU节点的数据总量占全网总数据量的比例：

$$CR = \frac{\sum_{i=1}^{N} c_i \cdot L_i}{\sum_{i=1}^{N} L_i} \tag{6}$$

其中，L_i 是RSU节点i的数据负载，CR 的值域为$[0,1]$。由于数据量大的热点区域被赋予了更高的权重，数据覆盖率的设计能够比单纯的节点覆盖率更能反映系统的实际服务效能。

2.3　工作负载模型

仅考虑最大化覆盖率可能导致边缘负载高度集中，从而引发边缘节点过载、服务排队增长与服务不稳定等问题。因此，均衡的负载分布对于防止单个边缘计算服务器过载，保障服务稳定性和延长网络寿命至关重要。因此，本文构建了工作负载模型，并描述了数据流量在已部署的边缘计算服务器之间的分布情况。在实际车联网部署过程中，边缘计算服务器的位置通常不会随着车辆移动而频繁调整，而是依据历史交通流量统计信息、区域业务需求以及长期运行数据进行规划部署[26-28]。因此，RSU负载并非表示某一时刻的瞬时负载，而是统计时间窗口内的平均业务需求，用于刻画区域长期计算服务需求分布。通过这种建模方式，车辆移动性和短时间尺度的业务波动已经通过统计负载的形式被反映，从而能够用于评估边缘计算服务器部署位置对整体服务能力的影响[29]。

对于部署$\left|S_{es}\right|$个边缘计算服务器的网络，RSU节点i的负载分配采用最近邻关联策略，这意味着产生的数据负载 L_i 将被卸载到距离其最近的边缘计算服务器。首先，需要找到离RSU节点i最近的边缘计算服务器：

$$j_i^* = \arg\min_{j \in S_{es}} d(v_i,v_j) \tag{7}$$

如果RSU节点i与其最近边缘计算服务器的距离满足覆盖条件$d(v_i,v_{j_i^*}) \leqslant R_{\max}$，则该RSU节点的整个数据负载 L_i 会被卸载给该边缘计算服务器。相反，数据负载 L_i 会被传输至云服务器。因此，第k个边缘计算服务器所承载的总工作负载计算如下：

$$W_k = \sum_{i=1}^{N} L_i \cdot \mathbb{I}\left(j_i^* = k \text{ and } d(v_i,v_k) \leqslant R_{\max}\right) \tag{8}$$

其中，$j_i^* = k$ 表示最近的边缘计算服务器是k，公式(8)通过对所有RSU进行遍历来获取被k覆盖的RSU总数据负载。为了量化所有边缘计算服务器之间工作负载分布的均匀性，采用负载标准差作为负载均衡度量（Workload Balance, WB）：

$$WB = \sqrt{\frac{1}{\left|S_{es}\right|}\sum_{k=1}^{\left|S_{es}\right|}(W_k - \bar{W})^2} \tag{9}$$

$$\bar{W} = \frac{1}{\left|S_{es}\right|}\sum_{k=1}^{\left|S_{es}\right|} W_k \tag{10}$$

其中，$\bar{W}$ 是所有边缘计算服务器的平均负载。WB 值越小，表明各边缘计算服务器负载越接近且系统负载越均衡。

2.4　延迟模型

在V2X网络中，端到端延迟直接影响车辆协同与安全服务质量。本文采用延迟模型来评估数据从产生到处理完成所经历的总时间。在该延迟模型中，总延迟由传输延迟、传播延迟和处理延迟三部分组成。通过对被边缘计算服务器覆盖和未被覆盖的RSU进行区分建模，任务的处理模式可以分为边缘计算模式和云端回传模式，两者具有不同的延迟特性。

传输延迟是数据从RSU发送到边缘计算服务器或云服务器接收所需的串行传输时间，计算如下：

$$T_i^{trans} = \frac{L_i}{\lambda_{trans}} \tag{11}$$

其中，λ_{trans} 是传输速率。

传播延迟是信号从发送端传播到接收端所需的时间。但当RSU节点未被任何边缘计算服务器覆

盖时，任务必须回传至云端。由于物理距离较远，其通常被包含在较高的处理惩罚中。因此，传播延迟只有在RSU被边缘计算服务器覆盖时才计算：

$$T_i^{prop}=\begin{cases}\dfrac{d(v_i,v_{j_i^*})}{\lambda_P}, & \text{if } d(v_i,v_{j_i^*})\leqslant R_{\max}\\ 0, & \text{otherwise}\end{cases} \tag{12}$$

其中，λ_P 是无线信号的传播速度。

处理延迟是边缘计算服务器或者云服务器处理计算任务所花费的时间，计算如下：

$$T_i^{proc}=\begin{cases}\dfrac{L_i}{\lambda_{edge}}, & \text{if } d(v_i,v_{j_i^*})\leqslant R_{\max}\\ \dfrac{L_i}{\lambda_{cloud}+\lambda_{penalty}}, & \text{otherwise}\end{cases} \tag{13}$$

其中，λ_{edge} 是边缘计算服务器的计算处理速率，λ_{cloud} 和 $\lambda_{penalty}$ 分别是云服务器的计算处理速率和处理惩罚。通常云计算能力是高于边缘计算的，即 $\lambda_{edge}<\lambda_{cloud}$。然而，由于边缘计算避免了远程传输，其整体延迟往往更低。

数据负载的总延迟由上述三部分延迟构成：

$$T_i=T_i^{trans}+T_i^{prop}+T_i^{proc} \tag{14}$$

系统的整体延迟性能以平均服务延迟（Average Delay, AD）来衡量，即计算所有RSU节点完成任务的平均时间：

$$AD=\frac{1}{N}\sum_{i=1}^{N}T_i\times 1000 \tag{15}$$

2.5 优化问题和形式化描述

本文聚焦于边缘计算服务器的静态规划部署，即在特定地理区域内RSU的固定位置以及典型业务负载模式下（如高峰时段的平均数据流量），来获取最优的边缘计算服务器部署位置。因此，在有限的边缘计算服务器部署资源约束下，如何通过优化边缘计算服务器的部署位置来实现V2X网络的计算服务覆盖率最大化，负载分布均衡化和传输延迟最小化的协同优化是至关重要的。本文将边缘计算服务器部署问题形式化为一个多目标组合优化问题，即在给定的部署约束下找到最优的部署集合，使系统的整体性能最优。

部署方案 $B=(b_1,b_2,...,b_N)$ 由覆盖率、负载均衡度和传输延迟共同评价。首先，定义决策变量 b_i 为边缘计算服务器在RSU节点 i 上的部署状态：

$$b_i=\mathbb{I}(v_i\in S_{es}) \tag{16}$$

其中，b_i 表示RSU节点 i 当前是否已部署了边缘计算服务器。$b_i=1$ 表示RSU节点已经部署了边缘计算服务器。同时，部署方案需要满足边缘计算服务器部署数量的约束：

$$\sum_{i=1}^{N}b_i\leqslant\left|S_{es}\right|_{\max} \tag{17}$$

其次，为了更好地量化不同性能指标的影响，需要将其归一化。归一化负载均衡度和归一化延迟性能计算如下：

$$WB_n=\max\left(0,1-\frac{WB}{WB_{\max}}\right) \tag{18}$$

$$AD_n=\max\left(0,1-\frac{AD}{AD_{\max}}\right) \tag{19}$$

然后，定义综合得分为覆盖率、归一化负载均衡度和归一化延迟性能的加权组合，表示如下：

$$Score=\omega_1\cdot CR+\omega_2\cdot WB_n+\omega_3\cdot AD_n \tag{20}$$

其中，ω_1，ω_2 和 ω_3 分别是覆盖率、归一化负载均衡度和归一化延迟性能的权重系数，且满足 $\omega_1+\omega_2+\omega_3=1$。

最后，V2X网络中边缘计算服务器部署优化问题可以表述为如下约束优化问题：

$$\text{s.t.}\sum_{i=1}^{N}b_i\leqslant\left|S_{es}\right|_{\max} \tag{22}$$

$$b_i\in\{0,1\},\quad \forall i\in\{1,2,...,N\} \tag{23}$$

其中，$\left|S_{es}\right|_{\max}$ 是最大允许部署的边缘计算服务器数量，各目标的权重系数设置通常是覆盖率权重较高，负载和延迟次之。

3 具体方案

针对V2X场景下边缘计算服务器部署所面临的覆盖范围有限、负载分布不均以及服务延迟较高等核心问题，提出一种面向V2X场景的边缘计算服务器部署方案（dueling double deep Q-learning based on residual graph attention network, ResGAT-D3QN）。该方案通过融合改进的图注意力网络（graph attention network, GAT）与竞争双深度Q网络（dueling double deep Q-learning, D3QN）来构建智能决策框架，实现在有限部署资源前提下的最大化计算服务覆盖率、均衡边缘计算服务器负载分布以及降低端到端传输延迟的目标。该方案通过三个关键模块来协同实现边缘计算服务器的部署优化，包括环境建模与特征编码模块、改进的GAT模块

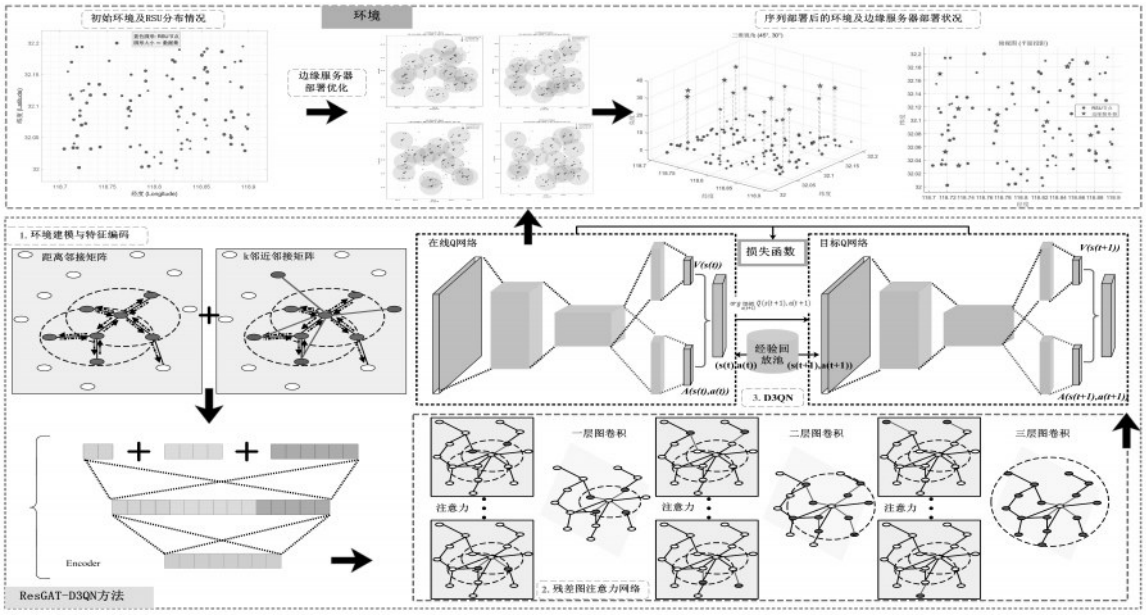


图4　基于残差图注意力网络的竞争双深度Q学习架构

$$\max_{B} \ Score = \omega_1 \cdot CR + \omega_2 \cdot WB_n + \omega_3 \cdot AD_n \quad (21)$$

以及基于D3QN的智能决策模块，方案架构如图4所示。

环境建模与特征编码模块用于获取V2X中RSU的地理分布等静态属性特征，边缘计算服务器的覆盖状态等动态部署特征以及节点重要性等综合决策特征来构建8维特征向量。其次，特征向量作为改进GAT模块的输入，通过图神经网络来学习RSU节点间的复杂依赖关系，为智能决策模块提供高质量特征支撑。最后，智能决策模块利用竞争双深度Q网络来学习边缘计算服务器的序列部署策略，通过多维度指标评估来量化部署方案的性能并通过设计的奖励函数来进一步优化部署策略。

3.1　环境建模与特征编码

V2X网络中的每个RSU都具有静态属性（如位置和数据量等）和动态状态（如是否已被选为部署点及当前覆盖情况），且RSU间的连接关系由地理距离和通信能力共同决定。然而，通用图建模方式难以精准表征网络特性。因此，环境建模与特征编码模块通过结合车路协同业务需求设计双重邻接矩阵，并构建多维融合节点特征向量，完成面向边缘部署场景的图环境定制化建模。具体来说，就是将这些异构信息编码为规范化的图结构 $G=(V,E,X)$ 并用于建模V2X网络部署环境。V 是RSU的集合，E 是边集合，X 是节点特征矩阵。这一过程包括两个关键环节，即基于双重策略的邻接矩阵 A 的构建以及包含8维信息的节点特征 x_i 的设计。

为了准确反映RSU节点间的关联性，邻接矩阵 A 的构建需要结合V2X特有的两个物理特性：（1）通信距离约束。由于无线通信信号的衰减特性，两个RSU间只有距离在一定范围内才能建立有效连接。（2）服务连续性需求。车辆在移动过程中需要在相邻RSU间无缝地切换服务。基于上述因素，提出了一种双重邻接矩阵构造策略来确保网络连通性和鲁棒性。

首先，需要根据通信半径 $R_{\max}$ 来构建基础的邻接矩阵 A_{dis}。对于有 N 个节点的 $V=\{v_1,v_2,\cdots,v_N\}$，节点 i 和节点 j 的距离邻接矩阵元素表示如下：

$$A_{dis}[i,j]=\begin{cases}1, & d(v_i,v_j)\le R_{\max}\\ 0, & \text{otherwise}\end{cases} \quad (24)$$

其中，$d(v_i,v_j)$ 是RSU节点 i 和节点 j 的距离，$A_{dis}[i,j]=1$ 表示节点 i 和节点 j 之间存在关联。

其次，为了避免基于通信距离构建的邻接矩阵可能导致的网络碎片化问题，这里还融入了 k 近邻连接模式（k-NN）来进一步增强拓扑表达能力。对于RSU节点 i，其 k 个邻居节点将会被选择与其连接且 k 值会受到网络规模的影响。节点 i 的邻近邻接矩阵元素表示如下：

$$A_{knn}[i,j]=\begin{cases}1, & v_j \in kNN(v_i,k)\\ 0, & \text{otherwise}\end{cases} \tag{25}$$

其中，当节点 j 属于节点 i 的 k 个邻居时，$A_{knn}[i,j]=1$。这里设置 $k=\min(8,N-1)$ 来确保图结构在稀疏分布的区域也不会退化为孤立点。

最后，通过结合两种邻接矩阵构建策略来获取完整地邻接矩阵：

$$A=\max(A_{dis},A_{knn}) \tag{26}$$

该双重邻接矩阵构造策略能够有效地平衡物理可行性与网络连通性。

节点特征的设计需要综合考虑信息的全面性和计算的有效性。因此，为了全面刻画每个RSU节点的状态，构建了一个8维的特征向量 $x_i \in \mathbb{R}^8$ 用于反映当前部署环境的关键信息。对于RSU节点 i，其8维特征向量定义如下：

$$x_i=[\tilde{lon}_i,\tilde{lat}_i,\tilde{L}_i,b_i,c_i,\tilde{d}_i,I_i,P] \tag{27}$$

其中，$\tilde{lon}_i$ 和 $\tilde{lat}_i$ 是RSU节点 i 的地理位置坐标，$\tilde{L}_i$ 是数据负载情况，b_i 表示边缘计算服务器的部署情况，c_i 表示边缘计算服务器的覆盖情况，$\tilde{d}_i$ 表示RSU节点 i 到边缘计算服务器的相对距离，I_i 是节点重要性指数，P 表示边缘计算服务器部署进度。

V2X场景中RSU节点 i 的静态属性特征包括经纬度坐标和数据负载量。经纬度坐标影响边缘计算服务器的覆盖区域，而数据负载量反映节点对于边缘计算服务的需求程度。归一化后的经纬度坐标和数据负载表示如下：

$$\tilde{lon}_i=\frac{lon_i-\mu_{lon}}{\sigma_{lon}+10^{-8}} \tag{28}$$

$$\tilde{lat}_i=\frac{lat_i-\mu_{lat}}{\sigma_{lat}+10^{-8}} \tag{29}$$

$$\tilde{L}_i=\frac{L_i-\mu_L}{\sigma_L+10^{-8}} \tag{30}$$

其中，σ_{lon} 和 μ_{lon} 分别是RSU经度的标准差和均值，μ_{lat} 和 σ_{lat} 分别是RSU纬度的均值和标准差，μ_L 和 σ_L 分别是数据负载的均值和标准差，L_i 是RSU节点 i 的数据负载量。

动态部署特征包括边缘计算服务器的部署情况 b_i，RSU被边缘计算服务器覆盖的情况 c_i 以及RSU到达边缘计算服务器的相对距离，这些特征随着边缘计算服务器部署更新而变化，具体表示如下：

$$\tilde{d}_i=\min\left(1,\frac{\min\limits_{j\in S_{es}} d(v_i,v_j)}{R_{\max}}\right) \tag{31}$$

其中，$\tilde{d}_i$ 表示RSU节点到最近的边缘计算服务器的相对距离。当RSU节点未被边缘计算服务器覆盖时，相对距离 $\tilde{d}_i=1$。

综合决策特征包括节点重要性指数和边缘计算服务器部署进度，这些特征考虑了数据负载情况，地理空间情况以及当前部署情况来进一步量化边缘计算服务器部署的优先级，从而辅助决策优化。其特征具体表示如下：

$$I_i=\beta\cdot DI_i+(1-\beta)\cdot CI_i \tag{32}$$

$$DI_i=\frac{L_i}{L_{\max}+10^{-8}} \tag{33}$$

$$CI_i=1-\frac{d(v_i,c)}{\max\limits_{k\in N} d(v_k,c)} \tag{34}$$

$$P=\frac{|\mathcal{S}_{es}|}{N_{\max}} \tag{35}$$

其中，节点重要性指数 I_i 是由数据重要性 DI_i 和地理重要性 CI_i 共同决定。c 是所有RSU节点的地理中心坐标。$d(v_i,c)$ 是RSU节点 i 与 c 的距离。RSU节点 i 的数据重要性由其相对数据负载占比表示，地理重要性则取决于其相对于地理中心位置的距离。权重系数设置为 $\beta=0.7$，来满足业务数据优先处理的需求，同时考虑地理位置的影响[30-32]。P 表示边缘计算服务器部署进度，该特征为全局共享特征并为决策感知提供帮助。这8维特征构建了节点特征矩阵 $X\in\mathbb{R}^{N\times 8}$，并结合邻接矩阵 A 来得到图结构数据，为图神经网络输入提供数据基础。

3.2　残差图注意力网络

从环境建模与特征编码模块获取图结构数据 (A,X) 后，需要从这些数据中学习并提取出影响部署决策的关键特征。然而，全连接网络难以捕捉V2X中RSU节点间空间分布关联性，如相邻节点存在协同服务需求。图神经网络是用于处理图结构数据的核心技术，但传统的图卷积神经网络难以满足V2X网络中RSU节点的异质性需求[33-34]。其原因在于图卷积神经网络赋予所有邻居节点相同的权

重，从而难以区分不同节点的重要性差异。此外，标准图注意力网络在层数加深后易出现梯度消失、节点特征过度平滑问题，无法满足RSU多跳空间依赖的提取需求。因此，本文引入残差连接与多重正则机制来构建增强型残差图注意力网络，从而适配 V2X 拓扑特征提取任务。残差图注意力网络通过注意力机制自适应学习不同节点间的权重来度量其关联程度。此外，通过结合残差连接和多层堆叠结构来构建深层图网络可以用于挖掘节点间的空间关联性和重要交互信息。最后，高维且异构的环境感知信息被转换为全局图级特征向量并用于为后续智能决策优化提供有效的支撑。

残差图注意力网络的核心思想是通过注意力机制为每个节点的邻居分配差异化的注意力权重，从而使节点特征能够正确反映拓扑结构中的关键信息。该网络架构包括特征输入层，图注意力的特征提取层和特征聚合层。具体如下：

（1）特征输入层：接收从环境建模与特征编码模块获取图结构数据 (A, X)，然后对输入特征进行维度变换，即将 8 维节点特征映射到隐藏层空间。

（2）特征提取层：通过图注意力层来自适应学习节点间的关联权重。通过一个可学习的权重矩阵 $\mathbf{W} \in \mathbb{R}^{8 \times d_h}$ 对每个节点的特征进行线性变换来得到新的特征表示作为第一层注意力层的输入：

$$H = \mathrm{ReLU}(X\mathbf{W} + \mathbf{b}) \tag{36}$$

其中，$\mathbf{W} \in \mathbb{R}^{8 \times d_h}$ 和 $\mathbf{b} \in \mathbb{R}^{d_h}$ 分别是可学习的投影矩阵和偏置参数，$d_h = 128$ 为隐层维度。ReLU 为激活函数，H 包含 $\{H_1, H_2, H_3, ..., H_N\}$，$H_N$ 为第 N 个节点的隐藏层特征。

为了量化节点 i 与节点 j 间的关联强度，需要计算节点间的注意力系数来衡量节点 j 对节点 i 的重要性，计算如下：

$$e_{ij} = \mathrm{LeakyReLU}\left(\mathbf{a}^{\top}[H_i \| H_j]\right) \tag{37}$$

其中，$\mathbf{a} \in \mathbb{R}^{2d_h}$ 是可学习的注意力权重向量。$\|$ 表示拼接操作，将 $H_i \in \mathbb{R}^{d_h}$ 和 $H_j \in \mathbb{R}^{d_h}$ 拼接为 $\mathbb{R}^{2d_h}$ 维向量，LeakyReLU的负斜率被设置为0.2来避免梯度消失。通过计算注意力分数，可以得到节点 j 对节点 i 的特征贡献重要程度。

此外，采用邻接节点极大负值掩码来更新注意力分数：

$$e'_{ij} = \begin{cases} e_{ij} & \text{if } A_{ij} > 0 \\ -\infty & \text{otherwise} \end{cases} \tag{38}$$

通过这样的设置可以让网络仅关注直接相连的节点，同时也减少了计算复杂度。然后，采用 softmax 函数对每个节点的邻居注意力进行归一化，来确保节点的所有邻居权重和为1：

$$\alpha_{ij} = \frac{\exp(e'_{ij})}{\sum_{k \in Ne(i)} \exp(e'_{ik}) + 10^{-8}} \tag{39}$$

其中，$Ne(i)$ 是节点 i 的邻居节点的集合，归一化后的注意力权重 α_{ij} 能够有效刻画邻居节点 j 对节点 i 的相对重要性，从而解决了传统图卷积操作所忽视的权重差异化问题。

（3）特征聚合层：对深层节点特征进行全局聚合并生成图级特征表示。具体来说，基于归一化的注意力权重，对节点 i 的所有邻居特征进行加权聚合来得到节点 i 的更新特征：

$$H'_i = \mathrm{ReLU}\left(\sum_{j \in Ne(i)} \alpha_{ij} \mathbf{W}_a H_j\right) \tag{40}$$

其中，$\mathbf{W}_a \in \mathbb{R}^{d_h \times d_h}$ 是可学习的变换矩阵。

为了捕获拓扑图中的多跳依赖关系，这里堆叠了多层图注意力层，并通过层的堆叠实现信息的深度传播。此外，为了防止深层网络出现的梯度消失或过度平滑问题，在每一层后引入了残差连接和层归一化。第 l 层图注意力网络的计算如下：

$$H^l_{mes} = \mathrm{GATLayer}^l(H^{l-1}, A) \tag{41}$$

$$H^l_{norm} = \mathrm{LayerNorm}(H^l_{mes}) \tag{42}$$

$$H^l_{act} = \mathrm{ReLU}(H^l_{norm}) \tag{43}$$

$$H^l_{drop} = \mathrm{Dropout}(H^l_{act}, p = 0.1) \tag{44}$$

$$H^l = H^{l-1} + H^l_{drop} \tag{45}$$

其中，$\mathrm{GATLayer}^l(\cdot)$ 为第 l 层图注意力层，输入为前一层的输出特征 H^{l-1} 和邻接矩阵 A，输出为当前层注意力聚合后的特征。$\mathrm{LayerNorm}(\cdot)$ 表示层归一化操作，即对特征的每个维度进行归一化来加速训练收敛并提高数值稳定性。$\mathrm{Dropout}(\cdot)$ 为丢弃层并以 $p = 0.1$ 的概率随机将部分特征置零来提供正则化效果防止过拟合。残差连接将输入特征与变换后的特征相加来确保网络在深层次时仍能保留低层的局部信息，从而能够有效缓解过度平滑。

节点特征通过残差图注意力网络已经融合了拓扑关联和深层语义信息。然而，部署决策需要基于整个网络的状态进行评估，并不单单是图注意力网

络输出的节点级特征矩阵。因此，需要将节点级特征聚合为图级表示。这里采用结合平均池化和最大池化的双通道池化策略来兼顾整体分布特征和关键节点突出特征：

$$G_{mean} = \frac{1}{N}\sum_{i=1}^{N} H_i^l \tag{46}$$

$$G_{max} = \max_{i=1,\ldots,N} H_i^l \tag{47}$$

$$G = [\, G_{mean} \,//\, G_{\max} \,] \tag{48}$$

其中，G_{mean} 为节点特征的均值，编码了网络的整体状态。平均池化对异常值不敏感并能够提供稳定的全局视图。G_{max} 为节点特征的最大值，用来捕捉关键节点的突出特征。二者拼接后的图级特征能够全面编码网络的状态信息。

3.3 竞争双深度Q决策框架

边缘计算服务器的部署是通过序列决策来实现的，即每一步都会从未部署的RSU中选择一个来部署边缘计算服务器，这需要在局部覆盖提升和全局性能优化之间权衡。然而，启发式算法存在的容易陷入局部最优问题、传统DQN存在的Q值过估计与决策稳定性不足的问题以及通用奖励函数无法适配多目标优化需求使得这些方法难以应用于实际的决策部署。因此，本文将边缘计算服务器的部署优化问题建模为马尔可夫决策过程，并采用竞争双深度Q网络来学习最优部署策略。其中，还增加了动态动作掩码机制约束决策空间，并设计了分层奖励函数引导多目标优化。该方法首先定义V2X决策部署场景下的状态空间、动作空间和奖励函数。然后通过价值函数分解、双网络更新以及经验回放机制来学习全局最优的序列部署策略，从而实现计算服务覆盖率、负载均衡和延迟降低的协同优化。具体如下：

（1）马尔可夫决策过程：为了精确描述V2X边缘计算服务器的部署问题，需要将其形式化为一个马尔可夫决策过程。其核心在于明确状态空间，动作空间和奖励函数，从而确保决策过程满足V2X场景下的约束条件和优化目标，即公式(21)-(23)。定义元组 $\mathbf{M} = (\mathbf{S},\mathbf{A},\mathbf{R})$，具体含义表示如下：

状态空间 **S**：由全局图嵌入 G 和节点特征矩阵 X 联合表示，具体定义如下：

$$\mathbf{S} = [\, G \,//\, X \,] \tag{49}$$

其中，t时刻的状态为 $s(t) \in \mathbf{S}$，由残差图注意力网络输出的全局图特征与节点特征拼接得到，且所有节点特征均经过归一化处理，取值被约束在有限区间。因此，状态空间 **S** 为有界有限集合，并全面反映了当前部署环境的关键信息，包括节点的静态属性、部署状态和网络拓扑等。

动作空间 **A**：是大小为N的离散动作空间，动作定义为选择一个未部署的RSU进行部署：

$$\mathbf{A} = \{\, i | i \in [\,0, N-1\,], i \notin S_{es} \,\} \tag{50}$$

动作 $a \in \mathbf{A}$ 意味着选择节点i部署边缘计算服务器。此外，引入了动态动作掩码 M_t 来确保每个RSU只能部署一台边缘计算服务器：

$$M_t = \begin{cases} 0, & \text{if } a \notin S_{es} \\ -\infty, & \text{if } a \in S_{es} \end{cases} \tag{51}$$

通过该设置，边缘计算服务器的部署只能在未被占用的RSU节点中进行选择，从而大大缩减了无效探索空间。由于受动态动作掩码约束，合法动作随部署进程动态缩减，因此动作空间为离散有限集合。

奖励函数 **R**：奖励函数是引导智能体学习最优策略的关键。在 t 时刻的即时奖励 $r(t) = \mathbf{R}(s(t),a(t),s(t+1))$ 可以量化单步决策的性能。基于公式(21)设计的多目标奖励函数表示如下：

$$r(t) = \lambda_1 r_d(t) + \lambda_2 r_s(t) + \lambda_3 r_c(t) + r_t(t) \tag{52}$$

$$r_d(t) = Score(t) - Score(t-1) \tag{53}$$

$$r_s(t) = Score(t) \tag{54}$$

$$r_c(t) = \max(CR(t) - CR(t-1), 0) \tag{55}$$

$$r_t(t) = \begin{cases} +10, & \text{if } CR(t) \geqslant Th1 \\ +5, & \text{if } Th1 \geqslant CR(t) \geqslant Th2 \\ +3, & \text{if } WB_{norm} \geqslant Th3 \end{cases} \tag{56}$$

其中，奖励函数 $r(t)$ 是由增量奖励 $r_d(t)$、核心奖励 $r_s(t)$、覆盖奖励 $r_c(t)$ 和终端奖励 $r_t(t)$ 共同构成。λ_1，λ_2 和 λ_3 分别是权重参数，$Th1$，$Th2$ 和 $Th3$ 分别是阈值。由于即时奖励由覆盖率、归一化负载均衡度和归一化延迟加权构成，三项指标值域均为[0,1]，且终端奖励仅在部署完成时给予。因此，奖励函数是有界的。其中，增量奖励用于奖励当前动作带来的性能提升，即鼓励每一步都尽可能的提升综合得分。核心奖励基于当前部署方案的综合性能来量化全局优化效果并确保智能体向高性能区域收敛。覆盖奖励用于额外的奖励覆盖率的直接

增长，进一步强化覆盖率提升的决策引导。终端奖励根据部署完成时的最终性能来给予额外奖励，从而进一步改善全局最优解。通过上述的奖励设置，不仅可以奖励最终结果，也可以奖励每一步的改善，从而有效缓解了稀疏奖励的问题。本文将V2X边缘计算服务器序列部署问题建模为有限MDP $\mathbf{M}=(\mathbf{S},\mathbf{A},\mathbf{R})$，满足经典Q-learning收敛的有限性、有界性假设[35]。此外，在采用DQN进行值函数近似时，由于图注意力网络输出的状态表征经归一化后分布稳定，结合残差连接与层归一化等机制，有效缓解了深度强化学习中的非平稳性问题，从而保障了训练过程的稳定性[33-34]。

（2）Dueling Q网络架构：相较于传统的DQN直接输出每个动作的Q值，Dueling Q网络能够将状态-动作函数 $Q(s,a)$ 分解为与动作无关的状态价值 $V(s)$ 和依赖于动作的优势函数 $A(s,a)$。这种分解方式能够使模型更准确的区分状态的好坏和动作的优劣，从而有效提升Q值估计的准确性。在车联网这种存在大量冗余状态的复杂环境中，该机制保证了即使在未充分探索所有动作分支的情况下，算法也能对状态本身的优劣做出准确的基线评估并高频更新状态价值 $V(s)$，从而在稀疏交互下依然保证了梯度的稳定传播和策略的快速收敛。网络的输入为状态向量，输出为N维的Q值向量，整体架构包括共享特征层，价值流分支，优势流分支和Q值融合[36-37]。

共享特征层：处理从残差图注意力网络获得的图级特征，并对其进行非线性变换来提取深层特征：

$$F'=\mathrm{Dropout}(\mathrm{ReLU}(s\mathbf{W}_1+\mathbf{b}_1),p=0.1) \quad (57)$$

$$F_s=\mathrm{ReLU}(F'\mathbf{W}_2+\mathbf{b}_2) \quad (58)$$

其中，F_s 是输出共享特征，$\mathbf{W}_1$ 和 $\mathbf{W}_2$ 分别是可学习的权重矩阵，$\mathbf{b}_1$ 和 $\mathbf{b}_2$ 为偏置项。Dropout层提供正则化，防止网络对特定特征过度依赖。共享特征层通过两次ReLU激活与Dropout操作增强了特征的非线性表达能力。

价值流分支：将共享特征映射为标量状态价值，其输出为全局状态价值 $V(s)$ 来反映当前状态的整体价值，表示如下：

$$V(s)=\mathrm{ReLU}\left(F_s\mathbf{W}_v+\mathbf{b}_v\right)\mathbf{W}_{v\_out}+\mathbf{b}_{v\_out} \quad (59)$$

其中，$V(s)$ 反映了当前网络状态的综合质量。$\mathbf{W}_v$ 和 $\mathbf{W}_{v\_out}$ 分别是降维权重矩阵和输出权重矩阵，$\mathbf{b}_v$ 和 $\mathbf{b}_{v\_out}$ 为偏置项。

优势流分支：并行处理共享特征并输出每个动作的优势值 $A(s,a)$ 用来反映动作 a 相对于其他动作的优势，表示如下：

$$A(s,a)=\mathrm{ReLU}(F_s\mathbf{W}_a+\mathbf{b}_a)\mathbf{W}_{a\_out}+\mathbf{b}_{a\_out} \quad (60)$$

其中，优势向量 $A(s,a)\in\mathbb{R}^N$ 的每个元素对应一个RSU部署动作的相对优势。$\mathbf{W}_a$ 和 $\mathbf{W}_{a\_out}$ 分别是降维权重矩阵和输出权重矩阵，$\mathbf{b}_a$ 和 $\mathbf{b}_{a\_out}$ 为偏置项。

Q值聚合：通过将状态价值和优势函数聚合得到最终Q值，计算如下：

$$Q(s,a)=V(s)+\left(A(s,a)-\frac{1}{N}\sum_{a'=1}^{N}A(s,a')\right) \quad (61)$$

聚合后的 $Q(s,a)$ 既能够反映当前状态的全局价值，又可以体现动作的相对优势，能够为部署决策提供准确的价值判断依据。

（3）Dueling Double DQN：为了缓解传统DQN中的Q值过估计问题，从而提升决策稳定性。本文采用了Double DQN的双网络机制，通过将动作选择和价值评估解耦并使用不同的网络参数，即在线网络参数 θ 和目标网络参数 θ^-。在线网络的参数是实时更新的并用于动作的选择和损失的计算。目标网络的参数是延迟更新的，主要用于计算目标Q值并避免同一网络的偏差积累。目标网络的参数更新采用软更新策略，通过逐步更新参数来确保稳定性。参数更新计算如下：

$$\theta^-\leftarrow\tau\theta+(1-\tau)\theta^- \quad (62)$$

其中，软更新系数 $\tau=0.01$ 使得目标网络的变化更加平滑，显著增强了训练的稳定性。此外，还需要构建经验回放池Buffer，表示如下：

$$\mathrm{Buffer}=(s(t),a(t),r(t),s(t+1),done) \quad (63)$$

其中，$s(t)$ 为当前的状态，$a(t)$ 为执行的动作，$r(t)$ 是获得的奖励，$s(t+1)$ 为下一时刻状态，$done$ 是终止标识，取值为1表示部署数量已经达到最大值，取值为0表示仍可以继续部署。经验回放池能够通过随机采样批量的样本来打破时间相关性使训练样本近似满足独立同分布假设，从而有效降低梯度更新的波动，可以用于网络参数更新和训练稳定性的提升，并为网络收敛奠定样本基础。

与标准DQN的目标计算 $y=r(t)+\gamma\cdot$

$\max\limits_{a(t+1)} Q(s(t+1),a(t+1);\theta^-)$ 相比，Double DQN 的目标Q值计算如下：

$$a_{t+1}^{*} = \arg\max_{a(t+1)} Q(s(t+1),a(t+1);\theta) \quad (64)$$

$$y = r(t) + \gamma \cdot (1-d) \cdot Q\left(s(t+1),a_{t+1}^{*};\theta^-\right) \quad (65)$$

其中，$\gamma \in [0,1]$ 为折扣因子，用于平衡即时奖励和未来奖励的权重。$(1-d)$ 确保终止状态无未来奖励以符合实际的部署过程。Double DQN 机制将动作的选择（由在线网络负责 θ）与动作的价值评估（由目标网络负责 θ^-）进行了解耦，有效阻断了正向误差的反馈循环，保障了价值函数估计的渐近无偏性。

Q学习的目标是最小化预测Q值与目标Q值之间的差异，损失函数的选择直接影响学习性能和收敛稳定性。本文采用Huber损失函数：

$$\text{SmoothL1}(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < \delta \\ \delta(|x| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \quad (66)$$

其中，δ 为切换阈值，$x = Q(s,a;\theta)$ 为时序差分误差。Huber损失结合了均方误差和平均绝对误差的优点。针对小误差，采用二次形式提供强梯度信号来加速收敛；针对大误差，采用线性形式减少异常值的影响来提高鲁棒性。这可以有效缓解探索初期环境高动态性带来的梯度爆炸，从而保障了网络训练的可靠性和网络权重的平稳收敛。总损失为批次内所有样本的Huber损失平均值：

$$\mathcal{L}(\theta) = \frac{1}{B}\sum_{i=1}^{B} \text{SmoothL1}\left(Q(s_i,a_i;\theta) - y_i\right) \quad (67)$$

此外，在选择动作时为了平衡探索新节点与利用已知最优解点，这里采用 ε - greedy 探索策略。该策略以概率 ε 选择合法动作，以概率 1 - ε 选择Q值最大动作，表示如下：

$$a(t) = \begin{cases} \text{Random}(A), & \text{with prob } \varepsilon \\ \arg\max\limits_{a \in A} Q(s(t),a), & \text{otherwise} \end{cases} \quad (68)$$

其中，ε 为探索率，并随着训练步数指数衰减：

$$\varepsilon(t+1) = \max(\varepsilon_{\text{end}}, \varepsilon(t) \cdot \varepsilon_{\text{decay}}) \quad (69)$$

其中，$\varepsilon_{\text{end}} = 0.05$ 用来保留少量探索避免局部最优，$\varepsilon_{\text{decay}} = 0.995$ 用于缓慢衰减并在探索与利用之间平衡。通过这样的设置，训练前期高探索率保证智能体充分遍历各类部署策略，避免陷入局部最优；训练后期探索率趋于稳定，算法从随机探索转向贪心利用最优动作，从而使得动作选择行为逐步平稳。

3.4 算法描述与复杂性分析

所提出的ResGAT-D3QN通过环境交互、经验存储、网络更新以及策略迭代来使智能体逐步学习到适配V2X场景下边缘计算服务器的最优部署策略。其训练过程包括初始化阶段，迭代训练阶段和训练终止与结果获取阶段，如算法1所示。

算法的1-2行展示了初始化阶段，包括数据的输入和参数的设置，初始竞争双深度Q网络的在线网络参数与目标网络参数是一致的[23-25]。算法的3-24行展示了迭代训练过程。在每轮训练中需要重置环境，将初始的特征矩阵 X 和邻接矩阵 A 输入到残差图注意力网络中生成图级特征 G，并构建初始状态 $\mathbf{S} = [G \,/\!/\, X]$。在每轮回合中，需要根据当前状态 $s(t)$ 选择动作 $a(t)$。然后执行动作并更新部署状态和计算即时奖励。更新后的 X 和 A 会生成新的图级特征并计算新的部署进度。接着将元组 $(a(t),s(t),r(t),s(t+1),done)$ 存入经验回放池Buffer中。当Buffer中的样本数量大于Batch Size时，从中随机抽取批量样本通过在线网络选择最优动作。计算目标网络的目标Q值和在线网络的损失 $\mathcal{L}(\theta)$，并更新在线网络的参数。目标网络参数通过软更新策略实现，并进行探索率衰减操作。最后，在完成 M 轮训练后输出最优的模型参数。

ResGAT-D3QN算法的整体时间复杂度由三个核心模块构成：环境建模与特征编码模块、残差图注意力模块、以及竞争双深度Q决策模块。首先，计算 N 个节点的双重邻接矩阵，需要获取节点间的距离，复杂度为 $O(N^2)$；其次，对每个节点进行特征向量编码，其复杂度为 $O(N)$。因此，环境建模与特征编码模块的时间复杂度为 $O(N^2)$。残差图注意力模块包含 L 层图注意力层，每层包含多头注意力机制和残差连接，单层GAT的复杂度为 $O(Nd_n^2 + kNd_n)$。因此，对于 L 层，其时间复杂度为 $O(LNd_n^2 + kLNd_n)$。竞争双深度Q网络包含在线网络和目标网络双分支，全连接层输入是ResGAT输出的全局特征。首先对已部署RSU节点执行掩码约束的复杂度为 $O(N)$；经过多层感知机输出 N 个动作的Q值，复杂度为 $O(Nd_n)$；然后从经验回

算法1：基于残差图注意力的竞争双深度Q学习方法的训练过程

输入：RSU数据集，最大训练轮数 M，每轮回合数 E，探索参数 ε，V2X环境Env

输出：训练的策略网络参数 θ

// 初始化阶段：

初始化环境并构建邻接矩阵 A，初始化在线网络参数 θ 和目标网络参数 θ^-，并设置 $\theta^- \leftarrow \theta$，初始化经验回放池Buffer

// 训练阶段：

For round $m = 1$ to M **do**:

　重置环境，$(A, X) \leftarrow$ Env.reset()

　获取图级特征 $G = \textbf{ResGAT}(A, X)$，并得到初始状态 $\mathbf{S} = [G \,//\, X]$

　For episode $e = 1$ to E **do**:

　　done $\leftarrow$ False

　　While not *done* **do**

　　获取有效动作集合 **A**

　　If Random() < ε **then**

　　以概率 ε 随机选择动作 $a \in \mathbf{A}$

　　Else

$$a(t) = \arg\max_{a \in A} Q(s(t), a)$$

　　// 环境交互与存储

　　执行动作 $a(t)$，$(r(t), s(t+1), done) \leftarrow$ Env.reset()

　　观测奖励 $r(t)$，新状态 $s(t+1)$ 和 *done*

　　将元组 $(a(t), s(t), r(t), s(t+1), done)$ 存入经验回放池Buffer中

　　If $|\text{Buffer}| \geqslant$ *Batch Size* **then**

　　从Buffer中采样一批数据，

　　$batch \leftarrow$ Buffer.sample(*Batch Size*)

　　计算目标值，损失并执行软更新

$$a^*_{t+1} = \arg\max_{a(t+1)} Q(s(t+1), a(t+1); \theta)$$

$$y = r(t) + \gamma \cdot (1 - d) \cdot Q\left(s(t+1), a^*_{t+1}; \theta^-\right)$$

$$\mathcal{L}(\theta) = \frac{1}{B}\sum_{i=1}^{B} \text{SmoothL1}\left(Q(s_i, a_i; \theta) - y_i\right)$$

$$\theta^- \leftarrow \tau\theta + (1 - \tau)\theta^-$$

　　$r(t), s(t), \mathbf{A}(t) \leftarrow r(t+1), s(t+1), \mathbf{A}(t+1)$

　　End While

　　探索率衰减 $\varepsilon(t+1) = \max(\varepsilon_{\text{end}}, \varepsilon(t) \cdot \varepsilon_{\text{decay}})$

　　End For

End For

return θ

放池中均匀随机采样批量大小 B 的样本并更新网络参数的复杂度为 $O(BNd_n)$。因此，经过 M 轮的训练，每轮训练 E 个回合，所提出算法的总时间复杂度为 $O(ME(N^2 + LNd_n^2 + kLNd_n + BNd_n)) = O(MEN^2)$。

4　实验结果

4.1　实验设置

实验采用模拟真实城市交通场景的RSU数据集[21]，包含100个RSU节点的完整信息，包括经纬度坐标、数据负载量等关键属性，其空间分布如图5所示。RSU的经纬度范围为经度118.70°-118.90°，纬度32.00°-32.20°。每个RSU的数据负载量根据其覆盖范围内的交通流量和业务需求设定，其中RSU节点数据负载量的不同决定其圆形标识大小。环境与实验参数设置如表2所示。

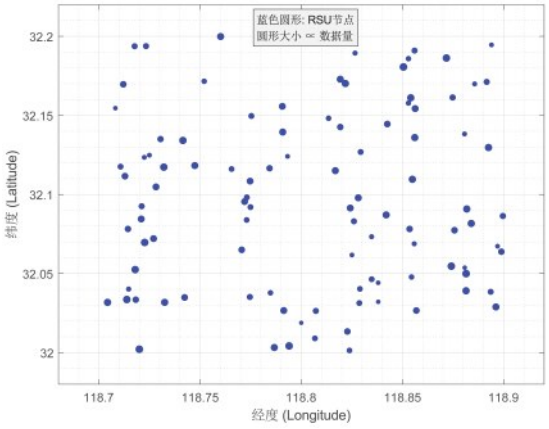


图5　RSU空间分布

表2　　**参数设置**

参数描述	值
RSU数量 N	100
边缘服务器数量 $\lvert\mathcal{S}_{\text{es}}\rvert$	15
覆盖半径 $R_{\max}$	3000 m
传输速率 λ_{trans}	1×10^7 bit/s
传播速度 λ_P	3×10^8 m/s
边缘计算速率 λ_{edge}	1×10^8 bit/s
云端计算速率 $\lambda_{cloud} + \lambda_{penalty}$	1×10^6 bit/s
学习率	3×10^{-4}
折扣因子 γ	0.99
批量大小	64
训练轮数 M, E	10, 1000

为了探索所提出的ResGAT-D3QN方法的性能优势，选取了以下三种方法进行对比实验：

1）ISC-QL方案[25]结合了改进的光谱聚类和Q-Learning算法来实现边缘计算服务器的部署。该方案在负载均衡度，能耗以及延迟方面提升显著。

2）WGTwo-Arch2方案[24]构建了多目标部署优化模型并采用生成对抗网络来实现RSU部署优化。结果表明，该方案能够有效解决RSU多目标部署优化问题。

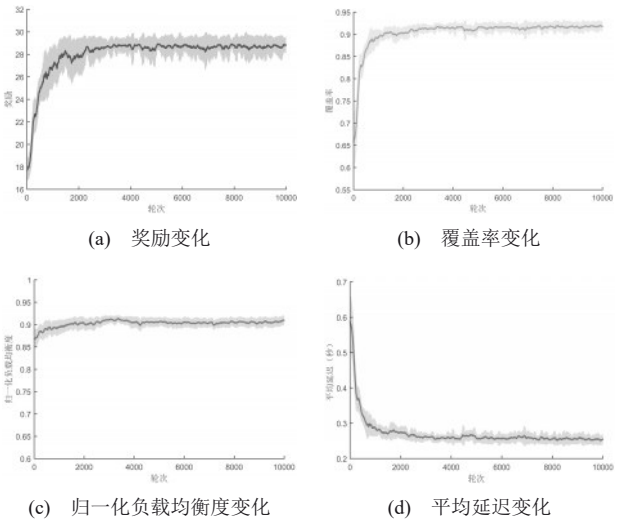
3）CKM-MAPPO方案[23]采用多智能体强化学习的方法来实现边缘计算服务器的部署和多目标优化问题的解决。结果表明，该方案能够有效改善负载均衡和传输延迟。

4.2　训练实验

训练实验旨在验证所提出的ResGAT-D3QN方法的收敛性和训练过程中的性能变化。实验记录了训练过程中奖励值，覆盖率，归一化负载均衡度和平均延迟的动态变化，如图6所示。

图6(a)展示了训练过程中奖励值的变化趋势。在训练初期，奖励值波动较大并快速上升，主要原因是智能体采用高探索率随机选择部署位置，部署策略缺乏针对性但可以积累多样化的经验数据。随着训练轮次的增加，探索率逐步衰减，智能体开始学习有效部署策略，奖励值呈现快速增长趋势，均值从20提升至28左右，这意味着ResGAT-D3QN可以有效地从图结构中提取特征同时逐渐掌握了哪些节点具备更高的部署价值，这使得每步的部署动作将带来综合性能的提升。在训练后期，奖励值趋于稳定，这得益于软更新策略有效的抑制了Q值的高估震荡，从而使智能体接近了最优的部署策略。此外，根据训练过程中平均奖励值变化趋势可以发现，随着训练轮数增加，奖励值逐渐提高并最终趋于稳定，表明所提出方法能够学习稳定的部署策略并实现稳定收敛。

图6(b)-(d)展示了训练过程中覆盖率，归一化负载均衡度和平均延迟的动态变化。三种指标的训练曲线呈现前期快速变化，中期平稳优化和后期稳步收敛的趋势。这一实验结果也证实了，在经验回放、衰减探索率、双网络软更新等机制共同作用下，ResGAT-D3QN算法能够迭代稳定并表现出良好的收敛性。从图6(b)-(d)可以观察到，覆盖率从初始的约0.65快速提升至0.92以上，并在6000轮



(a)　奖励变化　　(b)　覆盖率变化

(c)　归一化负载均衡度变化　　(d)　平均延迟变化

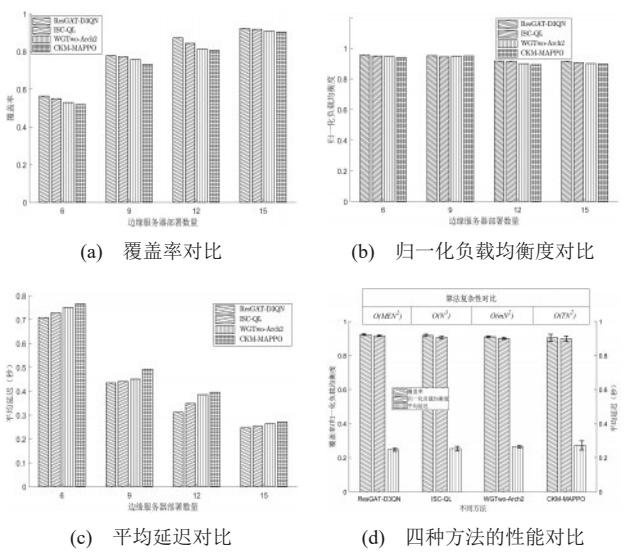
图6　训练过程中指标的变化情况

后基本稳定。这个阶段从开始的随机部署边缘计算服务器难以覆盖高负载热点区域逐步向优先选择高数据负载、地理位置关键的RSU部署边缘计算服务器以快速扩大有效覆盖范围的方向发展。归一化负载均衡度从0.86稳步上升至0.92左右。在训练初期，智能体优先追求覆盖率提升使得部分服务器负载过高，从而导致负载分布有待改善。在训练后期，智能体在选择部署位置时，不仅考虑覆盖的提升，还兼顾边缘计算服务器的负载状态分布。平均延迟从约600ms持续下降至240ms左右，原因在于覆盖率提升和负载均衡优化的协同作用：一方面，更多RSU被边缘计算服务器覆盖，避免了云端回传的高延迟；另一方面，负载均衡减少了排队等待时间并提升了处理效率。上述三种指标的改善得益于ResGAT-D3QN方法通过注意力机制动态聚焦重要节点及其邻域，使部署优先覆盖热点区域，从而快速提升覆盖率。此外，Dueling Double Q通过分离价值与优势估计来准确评估长期收益，从而引导智能体选择有助于降低整体延迟的部署动作。

4.3　对比实验

为了进一步探索所提出的ResGAT-D3QN方法的有效性，选取了三种部署方法在相同的RSU数据集和实验参数下分析不同边缘计算服务器部署数量对各性能指标的影响，如图7所示。

图7(a)展示了四种方法在不同部署数量下的覆盖率对比。随着边缘计算服务器部署数量增加，所有方法的覆盖率均呈上升趋势。此外，ResGAT-D3QN的覆盖率高于其他三种方法。WGTwo-Arch2



(a)　覆盖率对比　　(b)　归一化负载均衡度对比

(c)　平均延迟对比　　(d)　四种方法的性能对比

图 7　四种方法的性能指标

虽采用生成对抗网络，但未结合强化学习的序列决策能力，部署策略的动态调整能力受限。CKM-MAPPO的多智能体协作存在冗余决策问题也会影响覆盖效率。相反，ResGAT-D3QN通过残差图注意力网络提取深层拓扑特征，并结合Dueling Double DQN的序列决策优势能够优先选择高价值部署位置，从而在相同部署数量下实现更高的覆盖率。

图7(b)展示了归一化负载均衡度的对比结果。ResGAT-D3QN在各部署数量下均保持最高的负载均衡度，部署6台时负载均衡度为0.956，部署15台时达到0.916。这得益于ResGAT-D3QN使智能体在决策时同时考虑全局状态和局部动作影响并结合负载均衡相关的奖励设计，从而有效避免了负载集中并实现了更均衡的负载分布。然而，随着边缘计算服务器部署数量的增加，协同决策优化的难度也随之提升，使得四种方法的负载均衡度略有降低。

图7(c)展示了平均延迟对比。三种方法在部署15台边缘计算服务器时的平均延迟分别为0.253s，0.264s和0.272s。ISC-QL能够通过聚类算法和Q-Learning算法来动态优化部署位置以实现延迟的有效降低。WGTwo-Arch2和CKM-MAPPO虽在覆盖率和负载均衡上得到了优化，但其特征提取能力和决策精度不及ISC-QL，从而导致延迟优化效果受限。ResGAT-D3QN的平均延迟显著低于其他方法。原因在于ResGAT-D3QN通过图注意力网络学习RSU间的复杂依赖关系，优先选择能够实现更高覆盖率的部署位置，使得更少的任务需要回传至云端。同时，较好的负载均衡避免了单一服务器的排队拥塞，进一步降低了处理时延。

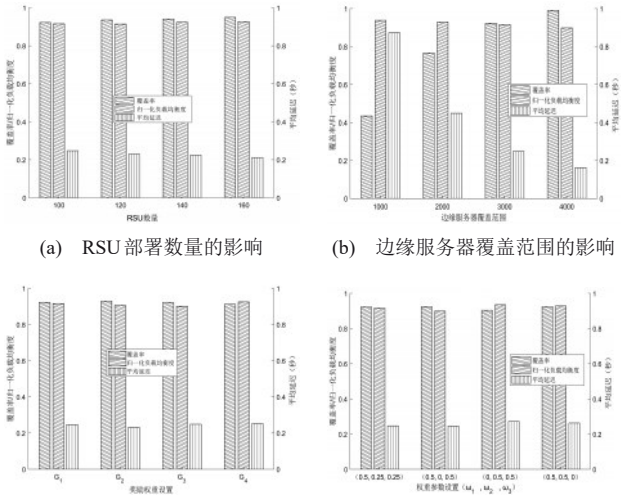
为验证算法稳定性，本文进行了10次独立重复实验，并统计各指标结果波动范围，如图7(d)所示。本文提出的ResGAT-D3QN在覆盖率、负载均衡度和平均延迟等指标上展现出了显著优势与较高的统计稳定性。原因在于所提出的方案能够通过全局图嵌入精准提取拓扑特征并结合Double Q机制有效抑制价值高估，从而在保障最优部署性能的同时将决策的随机波动限制在较低的范围内。但是ResGAT-D3QN中的残差图与竞争双深度Q网络也使得其有较高的算法复杂度。相比之下，ISC-QL方案与WGTwo-Arch2方案的平均延迟分别为0.253±0.014s和0.264±0.007s，其在综合性能上稍有逊色。由于ISC-QL方案需要进行特征分解，其算法复杂度最高 $O\left(N^3\right)$。CKM-MAPPO方案不仅在均值表现上相对较弱，其各项指标的波动范围也极为显著。但相较于其他方法，其算法复杂度相对较低。

4.4　参数影响实验

参数影响实验旨在分析关键参数变化对部署方案性能的影响，如图8至图9所示。

图8(a)探索了RSU数量的影响，随着网络规模增加，所提方法仍能够保持良好的覆盖性能、负载均衡能力和时延优化效果。如图8(a)所示，随着RSU数量的增加，覆盖率由0.923提升至0.950，同时平均延迟由0.247s下降至0.212s。这意味着更密集的RSU部署为边缘计算服务器提供了更多候选部署位置，有利于提高服务覆盖能力，从而降低了平均延迟。此外，归一化负载均衡度整体呈上升趋势，这表明所提方法能够在更大规模网络中实现更加均衡的资源分配。

图8(b)展示了边缘计算服务器覆盖范围的影响。随着覆盖半径的增大，覆盖率呈单调上升趋势。这是因为更大的覆盖半径能够让单个边缘计算服务器覆盖更多RSU节点，从而有效减少覆盖漏洞。然而，归一化负载均衡度呈现逐步下降的趋势，从0.938降低至0.898。原因在于，随着覆盖范围大幅增加，边缘计算服务器的覆盖区域会发生重叠。这导致处于数据热点的少数边缘计算服务器接收大量的数据，而处于边缘区域的服务器只覆盖到



(a)　RSU 部署数量的影响　　(b)　边缘服务器覆盖范围的影响

(c)　奖励函数设置的影响　　(d)　权重参数设置的影响

图 8　参数设置的影响

负载较低的区域。这种情况导致各边缘计算服务器间的负载标准差增大，从而使得归一化的负载均衡度下降。平均延迟随覆盖半径增大而单调下降，延迟从 0.873s 降至 0.162s。更大覆盖范围使边缘计算服务器覆盖更多 RSU 节点，从而降低云端回传导致的传输延迟和传播延迟。从上述结果中不难发现，盲目提升边缘计算服务器的覆盖范围来改善服务可用性和延迟的同时可能会损害资源的公平利用。

图 8(c)显示了奖励函数权重的影响。奖励函数权重被设置为四组，G_1 为初始设置 $\lambda_1 = 10$，$\lambda_2 = 1$ 和 $\lambda_3 = 5$，G_2 的 λ_1 =5，G_3 的 $\lambda_2 = 0.5$，G_4 的 $\lambda_3 = 2.5$。如图 8(c)所示，G_2 的服务覆盖率小幅提升至 0.929 且平均延迟降至 0.232s，但归一化负载均衡度略有下降至 0.906。这是因为增量奖励权重降低后，算法局部探索性增强，但对全局负载均衡的关注度有所减弱。G_3 的归一化负载均衡度略有下降至 0.901，其核心奖励权重降低导致智能体对综合得分的敏感度下降，进而影响整体优化效率。G_4 的覆盖奖励权重降低使服务覆盖率略有下降至 0.912，但归一化负载均衡度显著提升至 0.925，使负载均衡度得到了显著优化。

图 8(d)探索了综合得分权重系数的影响。这里选取四种权重组合：初始设置（0.5, 0.25, 0.25），侧重覆盖和负载均衡的设置（0.5, 0.5, 0），侧重覆盖和延迟的设置（0.5, 0, 0.5）以及侧重负载均衡和延迟的设置（0, 0.5, 0.5）。实验结果表明，奖励函数中权重的设置能够有效引导智能体的优化方向。对比四组结果，初始设置在三个指标上取得了较为均衡的表现。在达到 0.923 的高覆盖率和 0.247s 的低延迟的同时，也维持了 0.916 的高负载均衡度。这个结果表明，一个均衡且合理的权重设置有助于引导智能体探索到综合性能更优且更全面的部署方案，能够有效避免因过度优化某一指标而导致系统整体性能的下降。

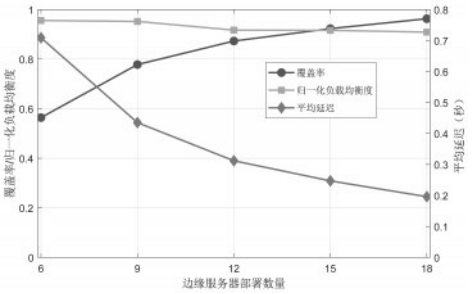


图 9　边缘计算服务器部署数量的影响

图 9 展示了边缘计算服务器最大部署数量对性能的影响。随着部署数量增加，覆盖率呈现增长趋势，而归一化负载均衡度出现下降趋势。这是因为负载分配采用最近邻关联策略，RSU 的负载只会分配给最近的边缘计算服务器，这使得边缘区域的服务器虽然数量增多，但其覆盖范围内可获取的负载增量有限。这种非对称的资源竞争加剧了各边缘计算服务器间的负载差异，进而使得归一化负载均衡度下降。平均延迟随着边缘计算服务器数量的增加而降低，即从 0.709s 降低至 0.196s。这个结果也证实了本方法能够在不同预算条件下寻找到符合多目标权衡的最优部署方案。

5　结束语

本文针对 V2X 网络中边缘计算服务器部署面临的服务覆盖受限、负载分布不均及端到端延迟高等问题，提出了结合残差图注意力网络与竞争双深度 Q 学习的智能部署方法 ResGAT-D3QN。该方法首先通过双重邻接矩阵与 8 维特征向量实现 RSU 网络的图结构建模。其次，利用残差图注意力网络捕获 RSU 节点之间的局部邻域关系和多跳空间依赖。最后，采用竞争双深度 Q 网络学习最优序列部署策略，并结合多目标奖励函数引导智能体在服务覆盖率、负载均衡与延迟之间实现协同优化。本文方案主要聚焦于静态边缘计算服务部署，但随着无人

机、移动基站和边缘车辆等新型边缘节点的引入，网络拓扑的动态性与服务资源的灵活性将进一步增强[38]。因此，未来的工作将通过静态与动态边缘计算服务器的协同部署与联合优化，全面考虑RSU负载的动态变化以及切换开销等关键因素，进一步提升边缘计算服务的灵活性与覆盖能力。

王烨，1994 年生，男。博士研究生，主要研究方向为物联网与边缘计算。

冉琼慧子，1999 年生，女。博士研究生。主要研究方向为区块链与车联网。

徐悦甦，1989 年生，男。博士，西安电子科技大学副教授。主要研究方向为软件工程，信息检索与人工智能。

高洪皓，1985 年生，男。博士，上海大学计算机学院教授、博导，长期从事软件智能研究，围绕工业互联网和具身智能开展创新应用

参考文献：

[1] 芮兰兰,邓淑予,陈子轩,等.基于多智能体深度强化学习的智能网联汽车服务迁移优化方法[J].通信学报,2026,47(01):141-155.

[2] Hassan M, Gupta S K, Mukhandi M, et al. Privacy in V2X communication: A comprehensive survey of threats, defenses, and emerging technologies[J]. Computer Science Review, 2026, 62: 100983.

[3] Liu B, Shi H, Jia D, et al. Collaborative Sensing and Communication for Intelligent Connected Vehicles: A Comprehensive Survey[J]. IEEE Communications Surveys & Tutorials, 2025, 28: 3125-3164.

[4] Abdi B, Mirzaei S, Adl M, et al. Advancing vulnerable road users safety: Interdisciplinary review on V2X communication and trajectory prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2025, 26(3): 2921-2943.

[5] 唐朝刚,李召,肖硕,等.一种面向车载边缘计算基于服务缓存的任务协同卸载算法[J].计算机学报,2025,48(04):864-876.

[6] Hakeem S A A, Kim H W. Advancing intrusion detection in V2X networks: A comprehensive survey on machine learning, federated learning, and edge AI for V2X security[J]. IEEE Transactions on Intelligent Transportation Systems, 2025, 26(8): 11137-11205.

[7] Liu G, Hu Y, Xu C, et al. Towards collaborative autonomous driving: Simulation platform and end-to-end system[J]. IEEE transactions on pattern analysis and machine intelligence, 2025, 47(8): 6566-6584.

[8] Yu H, Yang W, Zhong J, et al. End-to-end autonomous driving through V2X cooperation[C]//Proceedings of the AAAI conference on artificial intelligence. 2025, 39(9): 9598-9606.

[9] Hasan M K, Jahan N, Nazri M Z A, et al. Federated learning for computational offloading and resource management of vehicular edge computing in 6G-V2X network[J]. IEEE Transactions on Consumer Electronics, 2024, 70(1): 3827-3847.

[10] 李智勇,王琦,陈一凡,等.车辆边缘计算环境下任务卸载研究综述[J].计算机学报,2021,44(05):963-982.

[11] 许小龙,杨威,杨辰翊,等.车联网边缘计算环境下基于流量预测的高效任务卸载策略研究[J].电子学报,2025,53(02):329-343.

[12] 刘雪娇,曹天聪,夏莹杰.区块链架构下高效的车联网跨域数据安全共享研究[J].通信学报, 2023, 44(03):186-197.

[13] Wu H, Wang B, Ma H, et al. Collaborative caching relay algorithm based on recursive deep reinforcement learning in mobile vehicle edge network[J]. Ad Hoc Networks, 2024, 152: 103313.

[14] Hawlader F, Robinet F, Frank R. Leveraging the edge and cloud for V2X-based real-time object detection in autonomous driving[J]. Computer Communications, 2024, 213: 372-381.

[15] 赵庶旭,蒋恺俊,王小龙.边缘环境下面向移动性用户的微服务选择方法[J].通信学报,2025,46(05):200-217.

[16] Ning Z, Huang J, Wang X, et al. Mobile edge computing-enabled internet of vehicles: Toward energy-efficient scheduling[J]. IEEE Network, 2019, 33(5): 198-205.

[17] Wang S, Zhao Y, Xu J, et al. Edge server placement in mobile edge computing[J]. Journal of Parallel and Distributed Computing, 2019, 127: 160-168.

[18] Zhao X, Zeng Y, Ding H, et al. Optimize the placement of edge server between workload balancing and system delay in smart city[J]. Peer-to-Peer Networking and applications, 2021, 14(6): 3778-3792.

[19] Cao B, Fan S, Zhao J, et al. Large-scale many-objective deployment optimization of edge servers[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(6): 3841-3849.

[20] Chang L, Deng X, Pan J, et al. Edge server placement for vehicular ad hoc networks in metropolitans[J]. IEEE Internet of Things Journal, 2021, 9(2): 1575-1590.

[21] Lu J, Jiang J, Balasubramanian V, et al. Deep reinforcement learning-based multi-objective edge server placement in Internet of Vehicles[J]. Computer communications, 2022, 187: 172-180.

[22] Meng K, Liu Z, Xu X, et al. Heterogeneous edge service deployment for cyber physical social intelligence in internet of vehicles[J]. IEEE Transactions on Intelligent Vehicles, 2023.

[23] Zhou Z, Pooranian Z, Shojafar M, et al. An edge server deployment strategy for multi-objective optimization in the internet of vehicles[C]// 2024 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2024: 1-6.

[24] Fan S, Cao B. Many-Objective Edge Computing Server Deployment Optimization for Vehicle Road Cooperation[J]. Applied Sciences, 2025, 15(22): 12240.

[25] Zhou Z, Abawajy J. Reinforcement learning-based edge server place-

ment in the intelligent internet of vehicles environment[J]. IEEE Transactions on Intelligent Transportation Systems, 2025.

[26] Mazloomi A, Sami H, Bentahar J, et al. Reinforcement learning framework for server placement and workload allocation in multiaccess edge computing[J]. IEEE Internet of Things Journal, 2022, 10(2): 1376-1390.

[27] Li Z, Li G, Bilal M, et al. Blockchain-assisted server placement with elitist preserved genetic algorithm in edge computing[J]. IEEE Internet of Things Journal, 2023, 10(24): 21401-21409.

[28] Ling C, Feng Z, Xu L, et al. An edge server placement algorithm based on graph convolution network[J]. IEEE Transactions on Vehicular Technology, 2022, 72(4): 5224-5239.

[29] Taleb I, Guillaume J L, Duthil B. A survey on services placement algorithms in integrated cloud-fog/edge computing[J]. ACM Computing Surveys, 2025, 57(11): 1-36.

[30] Luo F, Zheng S, Ding W, et al. An edge server placement method based on reinforcement learning[J]. Entropy, 2022, 24(3): 317.

[31] Zhang Y, Lu Y, Yang G, et al. Multi-attribute decision making method for node importance metric in complex network[J]. Applied Sciences, 2022, 12(4): 1944.

[32] Yin Z, Bi Z, Xue J, et al. An improved heuristic service deployment algorithm based on multi-parameter priority in edge computing[J]. Cluster Computing, 2025, 28(7): 443.

[33] 吴涛,聂发志,先兴平,等.图基础模型研究进展与挑战:图神经网络的视角[J].通信学报,2025,46(07):226-248.

[34] 邹慧琪,史彬泽,宋凌云,等.基于图神经网络的复杂时空数据挖掘方法综述[J].软件学报,2025,36(04):1811-1843.

[35] Watkins C, Dayan P. Q-learning[J]. Machine learning, 1992, 8(3): 279-292.

[36] Xu C, Zhang P, Yu H, et al. D3QN-based multi-priority computation offloading for time-sensitive and interference-limited industrial wireless networks[J]. IEEE Transactions on Vehicular Technology, 2024, 73 (9): 13682-13693.

[37] Hu H, Wu D, Zhou F, et al. Intelligent resource allocation for edge-cloud collaborative networks: A hybrid DDPG-D3QN approach[J]. IEEE Transactions on Vehicular Technology, 2023, 72(8): 10696-10709.

[38] 尹浩,魏急波,赵海涛,等.面向有人/无人协同的智能通信与组网关键技术:现状与趋势[J].通信学报,2024,45(1):1-17.