Deep Reinforcement Learning for Resource Allocation in RIS-Assisted NOMA-MEC Vehicular Networks

Shunyao Wang[1], Wenjuan Yu[1], Chuan Heng Foh[2], Qiang Ni[1], Qiao Cheng[3], Lehu Wen[3]
[1]School of Computing and Communications, InfoLab21, Lancaster University, Lancaster, UK
[2]5GIC & 6GIC, Institute for Communication Systems (ICS), University of Surrey, Guildford, Surrey, UK
[3]Department of Electronic and Electrical Engineering, Brunel University London, Uxbridge, UK
Emails: wangsy2015ok@gmail.com, w.yu8@lancaster.ac.uk, c.foh@surrey.ac.uk,
q.ni@lancaster.ac.uk, {qiao.cheng, lehu.wen}@brunel.ac.uk

***Abstract*—Mobile edge computing (MEC) enables efficient computation offloading for mission-critical applications in resource-constrained vehicles, while reconfigurable intelligent surface (RIS) help address connectivity challenges for vehicles in urban environments with severe signal blockages. Non-orthogonal multiple access (NOMA) is an appealing technique that improves spectral efficiency while mitigating multi-user interference. This work proposes the RIS-assisted NOMA-MEC in vehicular networks, considering dynamic challenges such as heterogeneous vehicle processing capability, time-varying channel from high-mobility and dynamic task workloads. We formulate a system latency minimization problem by jointly optimizing the task offloading ratio, edge server resource allocation and RIS passive beamforming, while satisfying the task deadline and Signal to Interference plus Noise Ratio (SINR) requirements. To overcome the limitations of conventional optimization methods in such dynamic environments, we propose a soft actor critic (SAC)-based deep reinforcement learning (DRL) framework, which dynamically adapts to real-time channel state information (CSI), task workload and vehicle processing capability of all vehicles. Simulation results demonstrate that our approach achieves lower latency performance compared with the Deep Deterministic Policy Gradient (DDPG) baselines. Moreover, the proposed SAC method exhibits robustness and adaptivity to various levels of uncertainty in the CSI.**

***Index Terms*—Vehicle-to-everything (V2X), vehicular networks, MEC, RIS, NOMA, DRL.**

I. Introduction

Autonomous driving requires real-time sensing and environmental perception to maintain operational safety under dynamic and uncertain road conditions [1]. Such mission-critical applications demand substantial computational resources for real-time sensor data processing, posing significant challenges for vehicles with limited processing capability. To meet stringent computational demands, mobile edge computing (MEC) has been introduced to vehicular networks as a promising solution [2], [3]. MEC enables the offloading of computational tasks from resource-constrained vehicles to proximate edge servers, deployed at road side units (RSUs) or base stations (BSs). By leveraging the superior computation capability of edge servers, MEC can significantly reduce task processing latency for delay-sensitive tasks. In [4], the cloud-edge-vehicle computing architecture is established to reduce the task offloading latency. Moreover, in [5] and [6], unmanned aerial vehicles (UAVs) are deployed to assist MEC resource scheduling within vehicular networks.

In dense urban environments, the signals of vehicles are often blocked by various obstacles, such as high buildings, trees, and dense traffic flows. In such scenarios, reconfigurable intelligent surface (RIS) has emerged as a promising technology for next-generation wireless systems, providing energy-efficient dynamic control for signal propagation through passive beamforming [7]–[9]. RIS is a planar electromagnetic meta-surface composed of sub-wavelength passive elements, each capable of independently modulating the phase shift and amplitude of incident waves to reflect signals in desired directions. Consequently, RIS can establish virtual line-of-sight (LOS) propagation paths for vehicles under non-line-of-sight (NLOS) conditions, enhancing the received signal strength [7].

In addition to signal blockage commonly faced in urban environments, another practical challenge in vehicular networks comes from the high mobility of vehicles, which demands real-time optimization capable of adapting to highly dynamic and complex environments. Many studies have focused on static devices, failing to capture the critical issues introduced by vehicle mobility in dynamic environments [5], [6], [10], [11]. Moreover, conventional optimization methods are computationally intensive and difficult to support real-time optimization in rapidly changing environments [1]–[4], [6], [7]. For example, Zhang *et al.* [12] investigate signal-to-noise ratio (SNR) maximization for mobile vehicles but their work does not address mission-critical tasks in MEC. In this paper, we adopt deep reinforcement learning (DRL) as an effective solution for real-time optimization in such dynamic and complex scenarios [9]–[13].

Motivated by the aforementioned works, this paper investigates the DRL-based framework for the RIS-assisted NOMA-

This work was supported by HORIZON-MSCA-2022-SE-01 under the project TRACE-V2X (grant agreement ID 101131204).

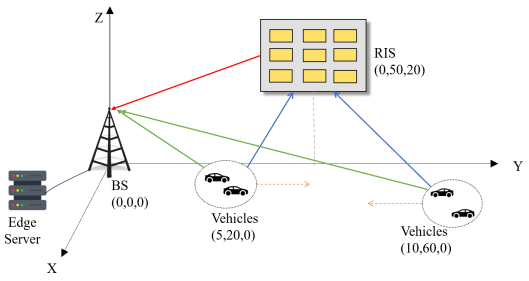

Fig. 1. The RIS-assisted NOMA-MEC vehicular Network.

MEC system in vehicular networks. The main contributions are summarized as follows:

- The proposed system considers three critical time-varying components in vehicular networks, including dynamic task load, time-varying wireless channels from high mobility, heterogeneous vehicle processing capability. This dynamic environment necessitates the real-time re-optimization at each time slot, which is a challenge for conventional optimization methods.
- A system latency minimization problem is formulated through jointly optimizing RIS beamforming, task offloading decision, and edge server resource allocation. The problem is modelled as a Markov Decision Process (MDP) and is solved using the soft actor-critic [14] (SAC) algorithm that dynamically adapts to real-time channel conditions, task workload, and heterogeneous vehicle processing capability.
- The SAC method shows about 15% reduction in latency compared to baseline algorithm in the 20-element RIS scenario. Moreover, the SAC method achieves near-optimal performance under low-uncertainty condition of 5% channel state information (CSI) error while maintaining performance stability under severe uncertainty of 25% CSI error through policy adaptation to channel variations.

II. System Model

A. Uplink Channel Model

As illustrated in Fig. 1, we consider a RIS-assisted NOMA-MEC system within a vehicular network, consisting of a base station (BS) integrated with an edge server, a RIS with N passive elements, and K vehicles ($\mathcal{K} = \{1, 2, \ldots, K\}$). Both BS and vehicles are equipped with single antenna. In this system, vehicles adopt uplink NOMA protocol for partial task offloading to the edge server through M orthogonal channels ($\mathcal{M} = \{1, 2, \ldots, M\}$). Each channel supports up to L vehicles to increase the spectral efficiency. We assume that the direct vehicles-BS link experiences severe signal attenuation due to environmental obstacles. Therefore, the RIS forms a cascaded vehicle-RIS-BS communication link, dynamically adjusting its phase shifts to direct signals from the vehicles to the BS.

The system operates in the discrete time slot $t \in \mathcal{T} = \{1, 2, \ldots, T\}$. We assume quasi-static channels that remain invariant within each time slot, but varying across slots. The perfect CSI is available at both BS and RIS. At time slot t, the RIS beamforming matrix is expressed as: $\boldsymbol{\psi}_t = \text{diag}\left(\sqrt{\beta_{1,t}}e^{j\theta_{1,t}}, \sqrt{\beta_{2,t}}e^{j\theta_{2,t}}, \ldots, \sqrt{\beta_{N,t}}e^{j\theta_{N,t}}\right)$, where $\beta_{n,t}$ and $\theta_{n,t}$ represent reflection amplitude and phase shift of the n-th element, respectively. We assume ideal reflection in this work, and thus $\beta_{n,t} = 1$ for all elements.

The direct channel from vehicle k to the BS is expressed as $\mathbf{h}_{k,B,t} \in \mathbb{C}^{1\times 1}$. The channel from vehicle k to RIS is expressed as $\mathbf{h}_{k,R,t} \in \mathbb{C}^{1\times N}$. The channel from RIS to the BS is given as $\mathbf{G}_{R,B,t} \in \mathbb{C}^{N\times 1}$. All channels follow Rician fading channel model, consisting of line-of-sight (LOS) and Non-line-of-sight (NLOS) components. The vehicle k-RIS link can be expressed as:

$$\mathbf{h}_{k,R,t} = \sqrt{PL_0 d_{k,R,t}^{-\alpha}}\left(\sqrt{\frac{\eta}{1+\eta}}\mathbf{h}_{k,R,t}^{\text{LOS}} + \sqrt{\frac{1}{1+\eta}}\mathbf{h}_{k,R,t}^{\text{NLOS}}\right), \tag{1}$$

where PL_0 is the reference pathloss at distance of 1 meter. $d_{k,R,t}$, α and η denote the time-varying distance between the vehicle k and the RIS, the pathloss exponent, and the Rician factor, respectively. The quantities $\mathbf{h}_{k,R,t}^{\text{LOS}}$ and $\mathbf{h}_{k,R,t}^{\text{NLOS}}$ indicate the LOS and NLOS components, respectively. In the RIS-assisted uplink transmission, vehicle k transmits the signal $x_{k,t}$ to the BS with $\mathbb{E}[|x_{k,t}|^2] = 1$. Therefore, the received signal at BS is:

$$y_t = \sum_{k=1}^{K}\sum_{m=1}^{M}\rho_{k,m,t}\sqrt{P_k}\mathbf{h}_{k,t}x_{k,t} + n_B, \tag{2}$$

where P_k is the transmit power of vehicle k. $\mathbf{h}_{k,t} = \mathbf{G}_{R,B,t}^H\boldsymbol{\psi}_t\mathbf{h}_{k,R,t} + \mathbf{h}_{k,B,t}$ is the composite channel, including the direct vehicle-BS link and the cascaded vehicle-RIS-BS link. $n_B \sim \mathcal{CN}(0, \sigma^2)$ is the Additive White Gaussian Noise (AWGN). $\rho_{k,m,t} \in \{0, 1\}$ denotes the channel allocation indicator, where $\rho_{k,m,t} = 1$ indicates that channel m is allocated to vehicle k at time slot t, and $\rho_{k,m,t} = 0$ otherwise. At time slot t, the achievable rate of vehicle k is given by:

$$R_{k,t} = \sum_{m=1}^{M}\frac{B}{M}\log_2\left(1 + \frac{\rho_{k,m,t}P_k|\mathbf{h}_{k,t}|^2}{\delta_{k,m,t} + \sigma^2}\right), \tag{3}$$

where $\delta_{k,m,t} = \sum_{\pi_m(k)\geq\pi_m(\hat{k})}\rho_{\hat{k},m,t}P_{\hat{k},t}|h_{\hat{k},t}|^2$, $\hat{k} \in \mathcal{K}\setminus\{k\}$ is the inter-user interference. $\pi_m(k)$ is the given decoding order of vehicle k on the mth channel. The BS employs successive interference cancellation (SIC) to decode superimposed signals from multiple vehicles.

B. Mobile Edge Computing

Assume that each time slot has a fixed duration of t_d. In time slot t, the vehicle k generates a computation task characterized by $\{D_{k,t}, F_{k,t}\}$, where $D_{k,t}$ is the task size (in bits) and $F_{k,t}$ is the vehicle's CPU frequency (in cycles/second). Using partial offloading, each vehicle offloads a portion of its task ($\alpha_{k,t}D_{k,t}$, with $\alpha_{k,t} \in [0, 1]$) to the edge server, while processing remaining part $(1 - \alpha_{k,t})D_{k,t}$ locally. The edge server allocates a fraction ($f_{k,t} \in [0, 1]$) of its computing resources to vehicle k. The edge server's computing resources

follow completeness of resource allocation constraint, i.e. $\sum_{k=1}^{K} f_{k,t} = 1$. The local computation time at vehicle k is $l_{k,t}^{\text{loc}} = \frac{(1-\alpha_{k,t})D_{k,t}C_{k,t}}{F_{k,t}}$, where $C_{k,t}$ represents the computing intensity (in CPU cycles/bit). The task offloading time from vehicle k to BS is $\tau_{k,t}^{\text{off,e}} = \frac{\alpha_{k,t}D_{k,t}}{R_{k,t}}$, where $R_{k,t}$ is the uplink data rate (Eq. (3)).

The edge server processes the offloaded task with allocated resource, leading to the task execution latency: $l_{k,t}^{e} = \frac{\alpha_{k,t}D_{k,t}C_e}{f_{k,t}F_e}$, where C_e and F_e represent server's computing intensity (in CPU cycles/bit) and total computation resources, respectively. Since the computation results are typically small in size, the downlink transmission time is neglected. Therefore, the total edge latency is $\tau_{k,t}^{e} = \tau_{k,t}^{\text{off,e}} + l_{k,t}^{e}$. The vehicle's task latency $L_{k,t}$ is determined by the slowest between local and edge latency: $L_{k,t} = \max(l_{k,t}^{\text{loc}}, \tau_{k,t}^{e})$. Assuming independent tasks, the server feedbacks the computation result immediately after completing each task, without waiting for other tasks. Thus, the total system latency is the worst-case among all vehicle latencies: $L_{\text{sys},t} = \max_{\mathcal{K}} L_{k,t}, \mathcal{K} = \{1, 2, \ldots, K\}$.

III. Problem Formulation

This paper aims to minimize the average long-term system latency in the RIS-assisted NOMA-MEC vehicular networks, through jointly optimizing the offloading ratio, edge computing frequency allocation, and RIS beamforming. The optimization problem is formulated as follows:

$$
\begin{aligned}
(\text{P1}) \quad & \min_{\alpha,f,\theta} \quad \frac{1}{T}\sum_{t=1}^{T} L_{\text{sys},t} && \text{(4a)}\\
& \text{s.t.} \quad 0 \le \alpha_{k,t} \le 1, \quad \forall k,t && \text{(4b)}\\
& \sum_{k=1}^{K} f_{k,t} = 1, 0 \le f_{k,t} \le 1, \quad \forall k,t && \text{(4c)}\\
& \theta_{n,t} \in [0, 2\pi], \quad \forall n \in \{1, \ldots, N\} && \text{(4d)}\\
& |e^{j\theta_{n,t}}| = 1, \quad \forall n \in \{1, \ldots, N\} && \text{(4e)}\\
& L_{\text{sys},t} \le t_d, \quad \forall k,t && \text{(4f)}\\
& \frac{\rho_{k,m,t}P_k|\mathbf{h}_{k,t}|^2}{\delta_{k,m,t} + \sigma^2} \ge \Gamma, \quad \forall k,m,t && \text{(4g)}
\end{aligned}
$$

The objective function (4a) minimizes the average latency over T time slots. Constraint (4b) specifies the offloading ratio range for each vehicle. Constraint (4c) ensures the completeness of resource allocation in edge server, while preventing over-allocation. Constraints (4d) and (4e) enforce RIS phase shift range and unit-modulus reflection requirements for passive beamforming. (4f) imposes the task deadline, requiring all tasks to be completed within time slot duration t_d. (4g) is the minimum SINR requirements for reliable NOMA operation.

Here we employ a simple channel allocation scheme for vehicles. All vehicles are sorted based on their channel gains and evenly divided into two groups: One with high channel gains and the other with low gains. Then, the NOMA clusters are formed by pairing high-gain and low-gain vehicles. Within each channel, the decoding order follows the descending order of vehicles' channel gains ($|h_{k,m,t}|^2$). The optimization of channel allocation and decoding order will be investigated in the future work.

The optimization of Problem (4) is challenging due to several inherent complexities. The first issue is the variable coupling effects. The offloading ratio α affects both communication latency and computation latency. Each phase shift element θ_n has coupled effects on all vehicular channels through beamforming. The second issue is the non-convex constraints. The RIS element unit-modulus constraint in (4e) creates a non-convex feasible set. In constraint (4f), fractional latency terms $l_{k,t}^{\text{e}}$ and $\tau_{k,t}^{\text{off,e}}$ introduce non-linearity. Thirdly, the objective function with max operator creates a non-differentiable surface, which makes the conventional convex optimization methods ineffective. Although some approaches can obtain sub-optimal solutions, they suffer from high computational complexity and intolerable execution time for practical deployments (e.g., $K > 20$ vehicles). To overcome these challenges, we develop a DRL framework to enable real-time optimization after offline training and deployment.

IV. DRL Framework for Optimization

In this section, we present the DRL-based framework to address the formulated optimization problem. Firstly, the problem (4) is modelled as a Markov Decision Process (MDP) with designed state, action and reward. Then, the SAC-based method is proposed to maximize the reward [14]. The SAC algorithm has several advantages over DDPG algorithm [15]. SAC generates the action probability distribution instead of deterministic outputs which means the agent learns both the mean performance and variance during the policy training process. Furthermore, by introducing an entropy term in Q value function, SAC can automatically balance exploration and exploitation. Besides, double critic networks are used to prevent the overestimation problem of Q value.

A. DRL Design

State space: The state space consists of four components: the task size for all vehicles $\mathbf{D}_t = [D_{1,t}, \ldots, D_{K,t}]$, vehicle CPU frequencies $\mathbf{F}_t = [F_{1,t}, \ldots, F_{K,t}]$, the cascaded vehicle-RIS-BS channel gains $\mathbf{H}_t^{\text{cas}} = [h_{1,t}^{\text{cas}}, \ldots, h_{k,t}^{\text{cas}}]$ with $h_{k,t}^{\text{cas}} = \mathbf{G}_{R,B,t}^{H}\boldsymbol{\psi}_t\mathbf{h}_{k,R,t}$, the direct vehicle-BS channel gains $\mathbf{H}_t = [h_{1,B,t}, \ldots, h_{k,B,t}]$. Thus, the state space is expressed as: $\mathbf{s}_t = \{\mathbf{D}_t, \mathbf{F}_t, \mathbf{H}_t^{\text{cas}}, \mathbf{H}_t\}$. The dimension of the state space is $4K$.

Action space: The action space consists of all optimization variables in problem (4): $a_t = \{\alpha_t, f_t, \theta_t\}$, where $\alpha_t = \{\alpha_{1,t}, \alpha_{2,t}, \ldots, \alpha_{K,t}\}$ represents the offloading ratio for all vehicles. $f_t = \{f_{1,t}, f_{2,t}, \ldots, f_{K,t}\}$ denote the edge server resource allocation. $\theta_t = \{\theta_{1,t}, \theta_{2,t}, \ldots, \theta_{N,t}\}$ indicates the phase shift of all RIS elements. Notably, the actions generated by SAC cannot be directly used for RIS-aided NOMA-MEC system. α_t and f_t are scaled into range $[0, 1]$ to satisfy constraints (4b) and (4c). The resource allocation vector f_t is processed through the softmax normalization

Algorithm 1 Update of Each Step in SAC Training Process

1: **Initialize:** Actor network π_{ϕ_a}, Critic networks Q_{ϕ_1}, Q_{ϕ_2}, Target networks $\tilde{Q}_{\tilde{\phi}_1}, \tilde{Q}_{\tilde{\phi}_2}$, Replay buffer $\mathcal{B}$
2: **Input:** mini-batch $\mathcal{D}$, entropy coefficient α_e, discount factor γ, learning rate l, target update rate τ
3: **Output:** offloading ratio α_t, edge resource allocation f_t, RIS phase shift θ_t
4: **for** episode=1 to $Episode_{max}$ **do**
5: Observe s_t from environment.
6: **for** step $t = 1, ...T$ **do**
7: Sample action a_t by actor's network via Eq. (6).
8: Execute a_t, get reward r_t and next state s_{t+1} from environment.
9: Store transition (s_t, a_t, r_t, s_{t+1}) in Buffer $\mathcal{B}$.
10: **if** $|\mathcal{B}| > |\mathcal{D}|$ **then**
11: Sample $\mathcal{D} = \{(s_d, a_d, r_d, s_{d+1})\}_{d=1}^{D} \sim \mathcal{B}$
12: Sample action a_{t+1} with s_{t+1} via Eq.(6)
13: **Critic Networks Update:**
14: Compute target Q-Value y_t via Eq.(9)
15: Compute Critic loss $\mathcal{L}(\phi_i)$ by Eq. (8)
16: Update Critic networks ϕ_i by Eq. (10)
17: **Actor Network Update:**
18: Compute policy loss $\mathcal{L}(\phi_a)$ by Eq. (12)
19: Update Actor network ϕ_a via Eq. (13)
20: **Target Network Soft Update:**
21: Update Target network $\tilde{\phi}_i$ via Eq. (11)
22: **Entropy Coefficient Adjustment:**
23: Compute temperature loss $\mathcal{L}(\alpha_e)$ via Eq. (14)
24: Adjust entropy coefficient α_e via Eq. (15)
25: **end if**
26: **end for**
27: **end for**

$\tilde{f}_{k,t} = \frac{e^{f_{k,t}}}{\sum_{k=1}^{K} e^{f_{k,t}}}$ to satisfy the completeness of resource allocation in constraint (4c): $\sum_{k=1}^{K} \tilde{f}_{k,t} = 1$. The phase shifts θ_t are scaled into range $[-\pi, \pi]$ instead of $[0, 2\pi]$ to align with the actor's tanh output, which maintains gradient continuity by eliminating discontinuous transition at $2\pi \to 0$ boundary. The total dimension of action space is $2K + N$.

Reward: Based on the current state s_t from environment, the SAC agent generates action set a_t, which is executed to obtain reward r_t. Since the DRL algorithm operates with ascending gradient to maximize cumulative rewards, the reward function is designed as the negative of objective function in (4a). To satisfy the task deadline (4f) and SINR constraint (4g), the reward function is defined as:

$$r_t = \begin{cases} -1 & \text{if (4f) or (4g) is violated,} \\ \max\left\{l_{k,t}^{\text{loc}}, \tau_{k,t}^{e}\right\} & \text{otherwise.} \end{cases} \tag{5}$$

B. Training Process

The update of each training step for the SAC method is shown in Algorithm 1. In each step, the actor decides the actions $(\alpha_t, f_t, \theta_t)$ based on the observed state s_t. All actions need to be processed as explained in the action space design above. Given the RIS phase shift θ_t, NOMA clustering are performed with the simple allocation scheme. After that, the system latency of all vehicles are obtained based on the decided offloading ratio α_t and the softmax-normalized edge CPU frequency allocation ratio $\tilde{f}_t$. After that, environment feedbacks the reward r_t and next state s_{t+1}. The transition (s_t, a_t, r_t, s_{t+1}) is stored in the replay buffer, which is used to update network. The details of the SAC updating framework are presented below.

SAC agent consists of an actor and two critic networks, each with a corresponding target network. The parameters of the actor network and critic networks are denoted by ϕ_a, ϕ_1 and ϕ_2, respectively. The target critic network parameters use $\tilde{\phi}_1$ and $\tilde{\phi}_2$, respectively. Given the observed state s_t, the action a_t is sampled from the Gaussian distribution generated by the actor network:

$$a_t \sim \pi_{\phi_a}(a_t|s_t) = \mathcal{N}(\mu_{\phi_a}(s_t), \sigma_{\phi_a}(s_t)), \tag{6}$$

where $\mu_\phi(\cdot)$ and $\sigma_\phi(\cdot)$ represent the mean and standard deviation (See line 7 in Alg. 1). The critic networks evaluate the actor's policy through estimating Q value $(Q_{\phi_i}(s_t, a_t), i = 1, 2)$, which is the expected long-term reward under current policy π_{ϕ_a}. SAC augments the Bellman function by adding an entropy term, defining the modified target Q function as:

$$Q_{\phi_i}(s_t, a_t) = r_t + \gamma \mathbb{E}_{s_{t+1} \sim P, a_{t+1} \sim \pi_{\phi_a}} \left[\min_{i=1,2} \tilde{Q}_{\tilde{\phi}_i}(s_{t+1}, a_{t+1}) - \alpha_e \log \pi_{\phi_a}(a_{t+1}|s_{t+1})\right], \tag{7}$$

where $\gamma \in [0, 1]$ is the discount factor, and α_e is the entropy temperature coefficient. the min operation represents the minimum Q-value from target networks. The entropy term $-\alpha_e \log \pi_{\phi_a}(a_{t+1}|s_{t+1})$ provides automatic exploration control and improved policy robustness against local optima. where α_e is the entropy weight to adjust the importance of the entropy. A large α_e increases the exploration and policy stochasticity, while a small α_e leads to the reward maximization.

During the training process, two critic networks update their parameters ϕ_1, ϕ_2 in parallel through the Bellman error minimization between the predicted Q value and target Q value (See lines 14-16 in Alg. 1). For each critic network (i=1,2), the loss function is:

$$\mathcal{L}(\phi_i) = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} (Q_{\phi_i}(s_t, a_t) - y_t)^2, \tag{8}$$

where $\mathcal{D} = (s_t, a_t, r_t, s_{t+1})_{d=1}^{D}$ is the mini-batch with size $|\mathcal{D}|$ sampled from experience replay buffer $\mathcal{B}$. The target Q value y_t is computed using the transition from buffer $\mathcal{B}$:

$$y_t = r_t(s_t, a_t) + \gamma[\min_{i=1,2} \tilde{Q}_{\tilde{\phi}_i}(s_{t+1}, a_{t+1}) - \alpha_e \log \pi_{\phi_a}(a_{t+1}|s_{t+1})], \tag{9}$$

where $\tilde{Q}_{\tilde{\phi}_i}$ denotes target Q networks. The min operator over two target Q values can mitigate the overestimation issue. Two

critic network update their parameters via the gradient descent of critic loss:

$$\phi_i \leftarrow \phi_i - l * \nabla_{\phi_i} \mathcal{L}(\phi_i), i = 1, 2, \tag{10}$$

where l is the learning rate for all networks.

To maintain the training stability, the target network parameters $\tilde{\phi}_i$ are softly updated with the critic parameters ϕ_i (See line 21 in Alg. 1):

$$\tilde{\phi}_i \leftarrow \tau\tilde{\phi}_i + (1-\tau)\phi_i, i = 1, 2, \tag{11}$$

where $\tau \in (0, 1)$ is the polyak averaging coefficient that controls the update rate.

The actor network updates its parameters ϕ_a by minimizing the policy loss (See lines 18-19 in Alg. 1):

$$\mathcal{L}(\phi_a) = -\frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}}(\min_{i=1,2} Q_{\phi_i}(s_t, a_t) - \alpha_e \log \pi_{\phi_a}(a_t|s_t)) \tag{12}$$

The gradient descent is performed as:

$$\phi_a \leftarrow \phi_a - l * \nabla_{\phi_a} \mathcal{L}(\phi_a), \tag{13}$$

where l and $\nabla_{\phi_a} J(\phi_a)$ are the step size and gradient of policy loss $\mathcal{L}(\phi_a)$.

Finally, the entropy temperature coefficient α_e is adaptively adjusted via minimizing the temperature loss (See lines 23-24 in Alg. 1):

$$\mathcal{L}(\alpha_e) = -\frac{\alpha_e}{|\mathcal{D}|}\sum_{\mathcal{D}}(\log \pi_{\phi_a}(a_t|s_t) + H_{target}), \tag{14}$$

where $H_{target} = -\dim(a_t)$ is the target entropy. The adjustment of α_e is:

$$\alpha_e \leftarrow \alpha_e - l * \nabla_{\alpha_e} \mathcal{L}(\alpha_e) \tag{15}$$

V. Numerical Results

In this section, the performance of the proposed algorithm for RIS-assisted NOMA-MEC systems is evaluated through numerical simulations. As shown in Fig. 1, we consider 3-D urban vehicular network scenario. The BS is located at (0, 0, 20) m, while the RIS is deployed at (0, 50, 20) m. Vehicles are distributed in two regions, near the BS and far from BS, with equal numbers in each region. All vehicles move in opposite directions along a bidirectional street with random speeds. Their positions are updated per time slot according to their speed. Additional simulation parameters are summarized in Table I.

Table I
Simulation Parameters

Parameters	Values
Maximum transmit power, $P_{k,\max}$	0.1 W
Noise power, σ^2	-97 dBm
Bandwidth, B	20 MHz
Path loss factor, α	2.2
SINR threshold, Γ	2
Time duration, t_d	100 ms
Vehicle speed	$[5, 10]$ m/s
Server computation resource, F_e	10 GHz
Computation intensity, C_k, C_e	550 Cycle/bit
Task size, D_k	[20-100] Kbit
Vehicular CPU frequency, F_k	[0.8,1] GHz
Discount factor, γ	0.99
Batch size	256
Buffer size	100000

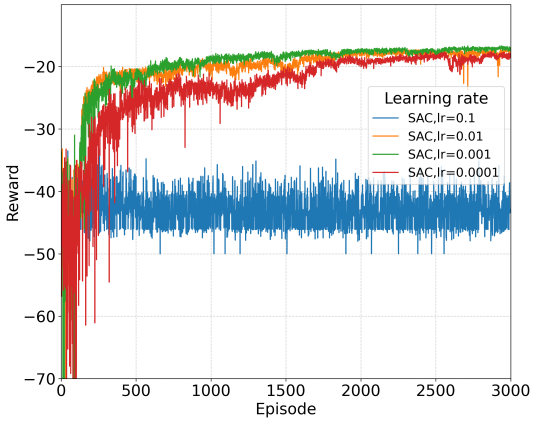

Fig. 2. The convergence performance of proposed method.

A. Convergence Performance of Proposed Algorithm

Fig. 2 demonstrates the convergence under different learning rates (lr). When the lr is set to 0.1, the SAC model fails to converge due to the gradient explosion in policy and critic networks, leading to random action outputs. Among different learning rates, lr = 0.0001 exhibits instability and the slowest convergence, which requires more training steps to converge. Although lr = 0.01 achieves rapid initial convergence, but shows unstable oscillations after episode 500. This prevents it converging to the optimal value. In contrast, lr = 0.001 shows slower initial convergence than lr = 0.01, but it keeps steady improvement and surpasses the performance of lr = 0.01 after episode 600. It indicates that a suitable learning rate can thoroughly explores the solution space. Based on results above, the most suitable learning rate for subsequent simulations should be lr = 0.001.

B. Impact of Imperfect CSI

This section investigates the impact of imperfect CSI on system latency. The wireless channels suffer from estimation inaccuracies due to vehicular mobility and practical CSI acuisition techniques. The imperfect CSI is modelled as Gaussian distributions: $\hat{\mathbf{H}}_t \sim \mathcal{CN}(\mathbf{H}_t, \sigma_{\text{CSI}}\mathbf{H}_t)$ and $\hat{\mathbf{H}}_t^{\text{cas}} \sim \mathcal{CN}(\mathbf{H}_t^{\text{cas}}, \sigma_{\text{CSI}}\mathbf{H}_t^{\text{cas}})$, with variances $\sigma_{\text{CSI}}\mathbf{H}_t$ and $\sigma_{\text{CSI}}\mathbf{H}_t^{\text{cas}}$.

Fig. 3 illustrates the convergence characteristics (average reward with corresponding variance) of the proposed algorithm under different learning rate (lr) and CSI uncertainty levels (σ_{CSI}). For lr = 0.001 with σ_{CSI} = 0.05, the model converges rapidly within 500 episodes, and steadily improves performance in subsequent training episodes. The SAC method shows robustness to low CSI inaccuracies. Under higher noise case (lr = 0.001 and σ_{CSI} = 0.25), although the convergence is slower with increased variance, the continuous performance improvement verifies its adaptivity to noisy environment.

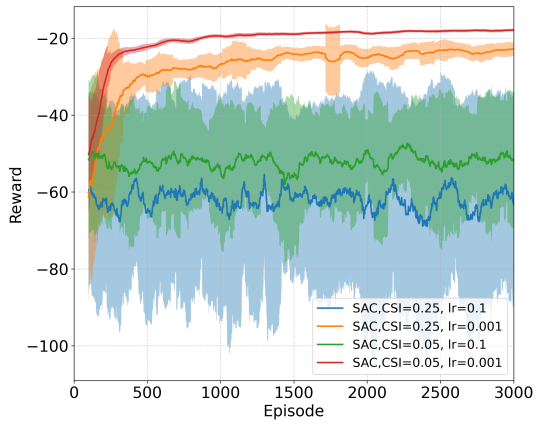

Fig. 3. The impact of CSI uncertainty over system latency.

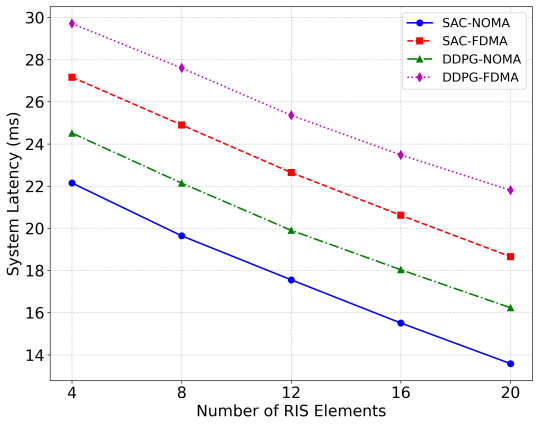

Fig. 4. The impact of different RIS elements on system latency.

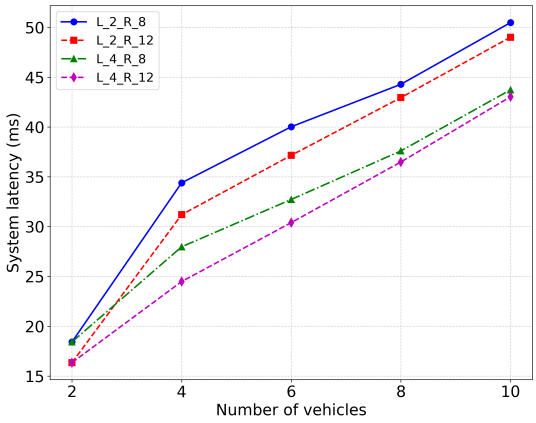

Fig. 5. The impact of number of vehicles on system latency.

Conversely, when using an unsuitable learning rate (lr = 0.1), the model diverges for both $\sigma_{\text{CSI}} = 0.05$ and 0.25, though the low-uncertainty case ($\sigma_{\text{CSI}} = 0.05$) shows comparatively better stability. The analysis above demonstrates that the proposed method is robust and adaptive to imperfect CSI environment under a suitable learning rate.

C. Impact of Number of RIS Elements

This section investigates the effect of different RIS element numbers on system latency. We compare four schemes: (1) SAC-NOMA: The proposed SAC-based method is combined with non-orthogonal multiple access (NOMA), in which two vehicles share a single channel (i.e., $K/2$ total channels); (2) SAC-FDMA: The SAC method is applied with frequency-division multiple access that allocates orthogonal channels to each vehicle (i.e., number of channels is K); (3) DDPG-NOMA. The conventional Deep Deterministic Policy Descent (DDPG) algorithm [15] is used with NOMA channel sharing; (4) DDPG-FDMA, combining DDPG with FDMA channel allocation. The evaluation considers both DRL algorithms (SAC and DDPG) and multiple access techniques (NOMA's channel sharing and FDMA's orthogonal channel allocation), which demonstrates the contributions of each component to system performance across different RIS elements.

Fig. 4 demonstrates the effects of varying RIS element numbers on system latency. As the number of RIS elements increases, the latency decreases across all schemes because of enhanced passive beamforming in the vehicles-RIS-BS cascaded channel. This confirms that larger RIS element numbers provide higher enhancement for uplink communication. Among the evaluated schemes, the SAC-NOMA scheme achieves the lowest latency, outperforming the other three schemes. DDPG-NOMA ranks the second in performance, while SAC-FDMA and DDPG-FDMA schemes exhibit higher latency. Notably, with 20 RIS elements, the SAC-NOMA scheme reduces system latency by 24% compared to SAC-FDMA. This result demonstrates that NOMA provides better spectral efficiency than FDMA. Additionally, our proposed SAC-based approach outperforms the DDPG in both NOMA and FDMA schemes. Particularly, with 20 RIS elements, the SAC-NOMA shows about 15% latency reduction compared with DDPG-NOMA scheme. The advantage come from SAC's double-critic architecture and entropy-regulated policy, which automatically balances the exploration and exploitation, leading to more stable training and faster convergence.

D. Impact of the Number of Vehicles

In this section, we investigate the impact of vehicle density on system latency. The proposed SAC-based method is evaluated under four configurations, combining different NOMA user capacities (L-2: 2 vehicles per channel, L-4: 2 vehicles per channel) and various RIS scales (R-8: 8 elements, R-12: 12 elements). In uplink NOMA, multiple users share the spectrum through Successive Interference Cancellation (SIC) decoding at BS. The BS subsequently decodes and removes the strongest signal from the composite received signal, before processing remaining signals. Therefore, four schemes are examined (L-2-R-8, L-2-R-12, L-4-R-8, and L-4-R-12) to evaluate their latency performance under different vehicle densities.

As shown in Fig. 5, when the testing with 2 vehicles, both L-2-R-8 and L-4-R-8 show identical latency, and the same as L-2-R-12 and L-4-R-12. This occurs because only two vehicles

utilize all bandwidth, regardless of NOMA channel capacity settings. The RIS configurations with same element (either R-8 or R-12) provide equivalent channel enhancement, resulting in the same latency for schemes with same number of RIS elements. Therefore, channel capacity of NOMA has no effect when the vehicle numbers equals to the minimum channel capacity (L-2).

For L-2 scenario, R-8 setup shows the near-linear latency growth, but deviates at 10 vehicles due to insufficient RIS elements. RIS with 8 elements cannot effectively enhance 10 vehicle channels. In contrast, the R-12 configuration maintains linearity, although its advantage diminishes with increasing vehicle density. For L-4 scenario (4 users per channel), both R-8 and R-12 maintain linear trends. L-4-R-8 performance outperforms L-2-R-8 due to higher spectral efficiency. The performance gap between R-8 and R-12 in L-4 scenario is narrower than in L-2 scenario, which proves that higher channel capacity reduces system sensitivity to RIS element numbers.

VI. Conclusion

This paper investigates joint communication-computation resource allocation in RIS-assisted NOMA-MEC vehicular networks. A SAC-based framework is proposed to minimize the system latency via jointly optimizing the task offloading ratio, edge server resource allocation and RIS beamforming. The framework dynamically adapts to environment challenges, such as time-varying task workloads, heterogeneous vehicle processing capability, dynamic channels and high-mobility of vehicles. The SAC-based method demonstrates the robustness and adaptivity to various CSI uncertainties. Besides, through simulations comparing with the DDPG baselines, our SAC-based method achieves lower average latency and faster convergence.

References

[1] Y.-D. Kim, G.-J. Son, C.-H. Song, and H.-K. Kim, "On the deployment and noise filtering of vehicular radar application for detection enhancement in roads and tunnels," *Sensors*, vol. 18, no. 3, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/3/837

[2] S. A. Ashraf, R. Blasco, H. Do, G. Fodor, C. Zhang, and W. Sun, "Supporting vehicle-to-everything services by 5g new radio release-16 systems," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 26–32, 2020.

[3] M. Cui, S. Zhong, B. Li, X. Chen, and K. Huang, "Offloading autonomous driving services via edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 535–10 547, 2020.

[4] Z. Xiao, X. Dai, H. Jiang, D. Wang, H. Chen, L. Yang, and F. Zeng, "Vehicular task offloading via heat-aware mec cooperation using game-theoretic method," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2038–2052, 2020.

[5] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu, and K. Yang, "Deep-learning-based joint resource scheduling algorithms for hybrid mec networks," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6252–6265, 2020.

[6] D. Van Huynh, Y. Li, A. Masaracchia, T. Hoang, and T. Q. Duong, "Optimal resource allocation for 6g uav-enabled mobile edge computing with mission-critical applications," in *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*, 2023, pp. 720–723.

[7] H. Luo, R. Liu, M. Li, and Q. Liu, "Ris-aided integrated sensing and communication: Joint beamforming and reflection design," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 7, pp. 9626–9630, 2023.

[8] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface-aided wireless communications: A tutorial," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3313–3351, 2021.

[9] Y. Eghbali, S. Faramarzi, S. K. Taskou, M. R. Mili, M. Rasti, and E. Hossain, "Beamforming for star-ris-aided integrated sensing and communication using meta drl," *IEEE Wireless Communications Letters*, vol. 13, no. 4, pp. 919–923, 2024.

[10] P. Qin, Y. Fu, Z. Yu, J. Zhang, and X. Zhao, "Urllc-aware trajectory plan and beamforming design for noma-aided uav integrated sensing, communication, and computation networks," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 1, pp. 1610–1625, 2025.

[11] J. Xie, "Deep q-learning aided energy-efficient caching and transmission for adaptive bitrate video streaming over dynamic cellular networks," *IEEE Access*, vol. 12, pp. 24 232–24 242, 2024.

[12] H. Zhang, R. Liu, M. Li, W. Wang, and Q. Liu, "Joint sensing and communication optimization in target-mounted stars-assisted vehicular networks: A madrl approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 7, pp. 10 011–10 025, 2024.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:205242740

[14] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1861–1870. [Online]. Available: https://proceedings.mlr.press/v80/haarnoja18b.html

[15] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2019. [Online]. Available: https://arxiv.org/abs/1509.02971