DRL-based Carbon Emission Optimization Method for the Vehicular Reverse Offloading System

Huijun Tang*, Chenguang Liu*, Jinjie Liu*, Hongjian Sun*, Pengfei Jiao† and Huaming Wu‡
*Department of Engineering, Durham University, Durham DH1 3LE, UK.
†School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
‡Center for Applied Mathematics, Tianjin University, Tianjin 300072, China
Emails: {huijun.tang, chenguang.liu, jinjie.liu, hongjian.sun}@durham.ac.uk, pjiao@hdu.edu.cn, whming@tju.edu.cn

***Abstract*—The rapid development of Intelligent Transportation Systems (ITS) and the Internet of Vehicles (IoV) has revolutionized transportation networks by enabling real-time communication between vehicles, road infrastructure, and cloud systems. One such advancement is the vehicular reverse offloading system, which allows Road Side Units (RSUs) to offload tasks to vehicles on the road. This paradigm makes full use of the dynamic computational resources available on vehicles and helps reduce the overall carbon emissions of IoV systems. In this paper, we establish a multi-hop reverse offloading vehicular edge computing model, enabling RSUs to utilize dynamic computational resources beyond their communication range. Considering the potential variations in power supply sources for RSUs, we further construct a carbon intensity adaptive carbon emission optimization model for RSUs and optimize the system's overall carbon emissions through deep reinforcement learning(DRL). Through extensive simulations, we demonstrate that our DRL-based approach significantly reduces carbon emissions compared to traditional task-offloading methods.**

***Index Terms*—Reverse offloading, Vehicular edge computing, Carbon emission, Deep Reinforcement Learning**

I. Introduction

The rapid development of Intelligent Transportation Systems (ITS) and the Internet of Vehicles (IoV) has revolutionized modern transportation networks by enabling real-time communication between vehicles, road infrastructure, and cloud systems [1]–[3]. With the growing demand for sustainability in intelligent transportation systems, optimizing carbon emissions has become a key challenge in vehicular communication networks. This interconnected environment provides opportunities for improved traffic management, safety, and efficiency. One such advancement is the vehicular reverse offloading system, where Road Side Units (RSUs) offload computational tasks to vehicles, which effectively utilizes the computational resources of vehicles, which are often underutilized, thus enhancing the overall network efficiency and resource utilization [4]–[6].

In addition to improving resource efficiency, the task offloading paradigm offers environmental benefits by reducing overall carbon emissions. Song *et al.* [7] propose a carbon emission optimization model in a Mobile Edge Computing(MEC) scenario. By employing an adaptive sensor node deployment strategy, it optimizes network channel allocation and minimizes carbon emission. Yang *et al.* [8] propose a carbon-aware dynamic task offloading online algorithm for the MEC system, which reduces carbon emissions while ensuring service latency. These approaches make task offloading decisions by considering the associated carbon costs, demonstrating that tasks can be offloaded in a carbon-efficient manner, thereby reducing the overall environmental carbon emissions of the MEC system.

Over the past few years, some approaches have been employed for the vehicular reverse offloading system. Gu *et al.* [5] propose a Deep Q-Network(DQN) based method to minimize the energy consumption and task delay of the vehicular reverse offloading system. Feng *et al.* [9] propose a joint alternative optimization-based bi-section searching for partial reverse offloading to minimize the latency in Vehicular Edge Computing(VEC) scenarios. These methods have demonstrated the potential for improving decision-making processes by dynamically adjusting reverse offloading strategies in VEC scenarios. Song *et al.* [10] propose a Deep Deterministic Policy Gradient(DDPG) based method to minimize the delay of the vehicular reverse offloading system. However, few studies have integrated carbon emission optimization within the vehicular edge offloading decision framework, and even fewer have considered the variability of power supply modes at RSUs.

The variability of power supply modes introduces significant challenges in energy consumption and carbon emissions in the offloading process [7], [11]. The energy supply to RSUs can vary depending on factors such as time of day, weather conditions, and the availability of renewable energy sources like hydropower, wind power, and solar energy [12]. Furthermore, the power supply mode changes can lead to situations where offloading decisions may increase carbon emissions if not correctly accounted for in the offloading decision-making process.

In this work, we consider a carbon emission optimization model for the vehicular reverse offloading system with the RSU and different power supply modes. Therefore, we propose a novel carbon emission optimization framework for multi-hop vehicular reverse offloading systems, which extends the communication and computational reach of the RSU by offloading tasks to vehicles beyond their direct communication range.

Corresponding author: Jinjie Liu.

979-8-3315-4444-7/25/$31.00 ©2025 IEEE

This framework incorporates a power supply mode-adaptive optimization algorithm based on DRL methods, which dynamically adjusts the offloading decisions based on the power supply mode of the RSU.

The main contributions of this paper are summarized as follows:

- We build a carbon emission optimization model for the multi-hop vehicle reverse offloading system, enabling efficient resource utilization and carbon emission reduction.
- We further propose a power supply mode-adaptive optimization algorithm based on deep reinforcement learning. Through deep neural networks, feature representations of input states are extracted, thereby achieving adaptive carbon emission optimization for different power supply modes of the RSU.
- We use real-world carbon intensity data to simulate the experiments. The results demonstrate that the proposed method outperforms the greedy-based approach, highlighting its effectiveness in optimizing carbon emissions.

II. System Model and Formulations

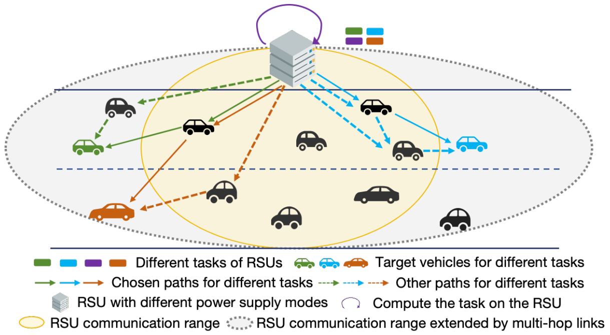

🟩 🟦 🟪 🟧 Different tasks of RSUs
Target vehicles for different tasks
Chosen paths for different tasks
Other paths for different tasks
RSU with different power supply modes
Compute the task on the RSU
RSU communication range
RSU communication range extended by multi-hop links

Fig. 1. The proposed vehicular reverse offloading system

In this paper, we consider a vehicular reverse offloading system, as illustrated in Fig. 1. The system consists of an RSU R and N vehicles with different speeds and directions, denoted by $V = \{V_0, V_1, \ldots, V_M\}$, where M is the number of the vehicles and V_0 represents the RSU. The speed of vehicle V_i is denoted by v_i, $i = 0, \ldots, M$. $v_0 = 0$ represents the RSU is stationary, a positive speed means the vehicle is moving to the right, and a negative speed indicates the vehicle is moving to the left. At each time slot t, the RSU generates a task $task_t^0$, $t = 1, \ldots, T$. For each task, we define w_t^0 as the required computational resources, D_t^0 as the task size, and ϕ_t^0 as the time constraint of the task of the RSU. In the proposed reverse offloading scenario, the RSU can choose to compute tasks locally or offload its tasks to vehicles on the road while the vehicles process their own tasks locally. We define the offload decision as $I_t = \{0, 1, \ldots, M\}$, where $I_t = 0$ means computing the task on the RSU and $I_t = i, i = 1, \ldots, M$ means computing the task the task on the i-th vehicle V_i.

However, among the possible offload choices, some are out of the RSU's communication range. For example, the green, orange, and blue vehicles in Fig. 1 are idle, but they are outside the communication range of the RSU. In the model we propose, the RSU can offload tasks to idle vehicles by utilizing other vehicles within its communication range to relay the data. To make full use of the computational resources available on the road, we utilize devices within the communication range as intermediaries to enable offloading to devices outside the communication range. Additionally, when the RSU transmits a task to a vehicle, the vehicle may already be occupied with its own tasks, thus causing a waiting delay in the offloading task. Therefore, carbon emissions are generated from three types of processes: transmission, waiting, and computing.

A. Transmission Process

The RSU and vehicles have limited communication ranges and cannot communicate with devices beyond their coverage areas. We denote $Cov = \{Cov_0, Cov_1, \ldots, Cov_M\}$ as the communication coverage set of the RSU and vehicles, where Cov_0 is the the communication coverage of the RSU.

In a vehicular network, the relative dynamics between two devices determine the duration of the communication link between them. We build a communication graph $G = (V, E)$, where E is the set of edges whose weights are the durations of the communication links, denoted as $T_{i,j}^{link}$, $i \neq j$ and $i, j = 0, \ldots, M$:

$$T_{V_i,V_j}^{link} = \frac{min(Cov_{V_i}, Cov_{V_j}) - Dis_{V_i,V_j}}{|v_i - v_j|}, \tag{1}$$

where $Dis_{i,j} = \sqrt{(y_i - y_j)^2 + (x_i - x_j)^2}$ is the distance between the devices V_i and V_j, whose positions are $loc_i = (x_i, y_i)$ and $loc_j = (x_j, y_j)$, respectively. $|v_i - v_j|$ is the relative speed between the devices V_i and V_j.

If $T_{i,j}^{link} > 0$, V_i and V_j can communicate. The transmission rate between V_i and V_j is as follows:

$$R_{V_i,V_j} = B_{V_i,V_j} log_2(1 + \frac{P_{V_i}^{tr} C}{\omega_0 Dis_{V_i,V_j}^{\vartheta}}), \tag{2}$$

where B_{V_i,V_j} is the bandwidth between two devices, $P_{V_i}^{tr}$ is the transmission power of the device V_i, C is the constant loss, ω_0 represents the noise power, and ϑ represents the path loss factor.

We define the set of vehicles within the RSU range as V^{in}, and the set of vehicles outside the RSU range as V^{out}.

1) 1-hop communication: : For $V_i \in V^{in}$, the transmission time is the one-hop transmission time of $task^{RSU}$ from the RSU to the vehicle V_i, which is as follows:

$$T_i^{tr_1} = \begin{cases} \frac{D^0}{R_{V_0,V_i}}, & \frac{D^0}{R_{V_0,V_i}} < T_{V_0,V_i}^{link} \\ +\infty, & \text{else} \end{cases} \tag{3}$$

where D^0 is the data size of the RSU and $R_{0,i}$ is the transmission rate from the RSU to the device V_i. $T_i^{tr_1} = +\infty$ represents the vehicle V_i drive out of the communication range before the task is transmitted. The carbon emission of the 1-hop communication process is as follows:

$$C_i^{tr_1} = \begin{cases} \frac{D^0}{R_{V_0,V_i}} P_0^{tr} C_0, & \frac{D^0}{R_{V_0,V_i}} < T_{V_0,V_i}^{link} \\ Carbon_c, & \text{else} \end{cases} \tag{4}$$

where P_0^{tr} is the transmission power of the RSU, C_0 is the carbon intensity of the RSU, and $Carbon_c$ is the carbon emission estimate value for offloading the task to the cloud. The cloud setting ensures that the task can still be successfully executed even when there is no feasible solution in the current vehicular reverse offloading system.

2) multi-hop communication: : For $V_i \in V^{out}$, we use multi-hop transmission. Suppose the communication between node V_0 and node V_i goes through multiple intermediate nodes, forming multi-hop paths $Path^i = \{Path_1^i, \dots, Path_{K_i}^i\}$, where K_i is the number of paths between the RSU and V_i. $Path_k^i$ is from the RSU through devices $V_1^{path_k^i}, \dots, V_{L_k^i}^{path_k^i}$ to the device V_i, where L_k^i is the length of the $Path_k^i$. For simplicity, we represent l_k^i as l and $V_l^{path_k^i}$ as V_l^k. We denote the duration of the k-th path $Path_k^i$ from the RSU to the device V_i as the bottleneck time $T_{i,k}^{tr_{bot}}$ of $Path_k^i$:

$$T_{i,k}^{bot} = \underset{l}{min}\{T_{V_l^k, V_{l+1}^k}^{link}, T_{V_L^k, V_i}^{link}\} \tag{5}$$

where $l = 0, 1, \dots, L-1$ and $l = 0$ represent the link is from the RSU.

Considering there are multiple paths between the RSU and the device V_i, we select the path with the longest transmission time among all paths $Path_k^i$ from the RSU to the vehicle V_i, $k = 1, \dots, K_i$, to transmit the task, which corresponds to the path with the maximum bottleneck time for communication. So the best path to device i among all path $Path_k^i$ is determined by the following formula:

$$k_i^* = \underset{k}{argmax}\{T_{i,k}^{bot}\}, \tag{6}$$

Transmitting $task^{RSU}$ from the RSU to the device V_i by the multi-hop paths costs $\hat{T}_{i,k^*}^{tr}$:

$$\hat{T}_{i,k^*}^{tr} = \sum_{l=0}^{L} \frac{D^0}{R_{V_l^{k^*}, V_{l+1}^{k^*}}} + \frac{D^0}{R_{V_L^{k^*}, V_i}}, \tag{7}$$

The carbon emission of when transmitting by the multi-hop path is as follows:

$$\begin{aligned} \hat{C}_{i,k^*}^{tr} = {} & \frac{D^0}{R_{V_0, V_1^{k^*}}} P_0^{tr} C_0 + \sum_{l=1}^{L} \frac{D^0}{R_{V_l^{k^*}, V_{l+1}^{k^*}}} P_v^{tr} C_{V_l^{k^*}} \\ & + \frac{D^0}{R_{V_L^{k^*}, V_i}} P_v^{tr} C_{V_L^{k^*}}, \end{aligned} \tag{8}$$

where P_v^{tr} is the transmission power of vehicles, $C_{V_l^{k^*}}$ and $C_{V_L^{k^*}}$ are the carbon intensity of vehicles $V_l^{k^*}$ and $V_L^{k^*}$, which corresponds to the relay sub-paths for multi-hop task offloading and the last path belong the chosen path k_i^*.

The transmission time for multi-hop paths from the RSU to the device V_i is as follows:

$$T_i^{tr_2} = \begin{cases} \hat{T}_{i,k^*}^{tr}, & \hat{T}_{i,k^*}^{tr} < T_{i,k^*}^{bot} \\ +\infty, & \text{else} \end{cases} \tag{9}$$

The actual carbon emission for transmission process along multi-hop paths from the RSU to the device V_i is as follows:

$$C_i^{tr_2} = \begin{cases} \hat{C}_{i,k^*}^{tr}, & \hat{T}_{i,k^*}^{tr} < T_{i,k^*}^{bot} \\ Carbon_c, & \text{else} \end{cases} \tag{10}$$

So the transmission time for the offloading decision I_t is as follows:

$$T^{tr}(I_t) = \begin{cases} T_{I_t}^{tr_1}, & \text{if } I_t \in V^{in,t} \\ T_{I_t}^{tr_2}, & \text{if } I_t \in V^{out,t} \end{cases} \tag{11}$$

where $V^{in,t}$ and $V^{out,t}$ are the vehicle sets for vehicles in the RSU communication range and out of the RSU communication range in time slot t, respectively.

The carbon emission in the transmission process for the offloading decision I_t is as follows:

$$C^{tr}(I_t) = \begin{cases} C_{I_t}^{tr_1}, & \text{if } I_t \in V^{in,t} \\ C_{I_t}^{tr_2}, & \text{if } I_t \in V^{out,t} \end{cases} \tag{12}$$

B. Waiting Process

At time $t = 0$, the initial queues of the RSU and vehicles are set as $\Gamma_0^0 = w_0^0$ and $\Gamma_0^i = w_0^i$, where w_0^0 and w_0^i initial computational resource requirements for the tasks at the RSU and vehicles, respectively. Each device generates a task at each slot. If there are incomplete tasks on device V_i, the offloaded task needs to wait.

We define the indicator function $\mathbb{I}(I_t = i))$ to indicate whether the task is executed on device V_i. The queue of the RSU at $t+1$ is as follows:

$$\Gamma_0(t+1, I_t) = \max\left(0, \Gamma_0(t) + w_t^0 \cdot \mathbb{I}(I_t = 0) - F^0\right) \tag{13}$$

where $t = 0, 1, \dots, T-1$, $\Gamma_0(t)$ represents the queue at the RSU at t, w_t^0 is the computational resource requirements of the task generated by the RSU at time t, and $F^0 = f_0 \triangle t$ is the amount of the task the RSU can process within a slot. f_0 is the computational capability of the RSU, and $\triangle t$ is the length of a time slot.

The vehicle V_i prioritizes tasks that remain unfinished from the previous slot, followed by tasks generated locally in the current slot, and finally processes tasks offloaded from the RSU. Therefore, the queue of the vehicle V_i at $t+1$ is:

$$\Gamma_i(t+1, I_t) = \max\left(0, \Gamma_i(t) + w_{t+1}^i + w_t^0 \cdot \mathbb{I}(I_t = i) - F^i\right) \tag{14}$$

where $i = 1, \dots, M$, $t = 0, 1, \dots, T-1$, $\Gamma_i(t)$ is the queue of V_i at t, w_{t+1}^i is the computational resource requirements of the task generated by V_i at $t+1$, $w_t^0 \cdot \mathbb{I}(I_t = i)$ is the computational resource requirements of the task offloaded by the RSU to V_i at time t, and $F^i = f_i \triangle t$ is the amount of the task the vehicle V_i can process within a slot. f_i is the computational capability of V_i.

The wait time at t is:

$$T^W(I_t) = \sum_{i=0}^{M} \frac{\Gamma_i(t, I_t)}{f_i} \tag{15}$$

We compute the carbon emission at waiting process $C^W(I_t)$ as follows:

$$C^W(I_t) = \sum_{i=1}^{M} \frac{\Gamma_i(t, I_t)}{f_i} P_v^W C_{V_i} + \frac{\Gamma_0(t, I_t)}{f_0} P_0^W C_0 \tag{16}$$

where P_0^W and P_v^W are the idle power of the RSU and vehicles, respectively. C_{V_i} is the carbon intensity of V_i

C. Computing Process

The execution time $T^P(I_t)$ for the task of the RSU at t is:

$$T^E(I_t) = \sum_{i=0}^{M} \frac{w_t^0 \cdot \mathbb{I}(I_t = i)}{f_i} \tag{17}$$

The carbon emission for computing process is as follows:

$$C^E(I_t) = \sum_{i=1}^{M} \frac{w_t^0 \cdot \mathbb{I}(I_t = i)}{f_i} P_v^E C_{V_i} + \frac{w_t^0 \cdot \mathbb{I}(I_t = 0)}{f_i} P_0^E C_0 \tag{18}$$

where P_0^E and P_v^E are the execution power of the RSU and vehicles, respectively.

D. Optimization Model under Different Power Supply Modes

Different power supply modes have varying carbon intensities. For instance, power sourced from fossil fuels leads to higher carbon emissions, while renewable energy sources result in significantly lower emissions. By adjusting offloading decisions based on the carbon intensity of the available power mode, the system can effectively reduce its overall carbon emission.

In this paper, we consider a carbon emission optimization model for the vehicular reverse offloading system with the RSU that has different power supply modes. We define $\Lambda = \{\alpha_q \mid q \in \mathbb{N}, 1 \leq q \leq |\Lambda|\}$ as the set of different power supply modes, each of which is associated with a carbon intensity $C_0(\alpha_q)$. The carbon emissions in the transmission process under different power supply modes are as follows:

$$C_{\alpha_q}^{tr}(I_t) = \begin{cases} C_{I_t}^{tr_1}(C_0(\alpha_q)), & \text{if } I_t \in V^{in,t} \\ C_{I_t}^{tr_2}(C_0(\alpha_q)), & \text{if } I_t \in V^{out,t} \end{cases} \tag{19}$$

where $C_{I_t}^{tr_1}(C_0(\alpha_q))$ and $C_{I_t}^{tr_2}(C_0(\alpha_q))$ is computed by Eq (4) and Eq (10) where $C_0 = C_0(\alpha_q)$, respectively. Similarly, the carbon emission in waiting and computing process under different power supply modes are defined as $C_{\alpha_q}^W(I_t)$, $C_{\alpha_q}^E(I_t)$ computed by Eq (16) and Eq (18) where $C_0 = C_0(\alpha_q)$, respectively.

This model aims to minimize the total carbon emissions associated with the system's communication and computation processes. The formulation of this optimization problem can be expressed as follows:

$$C_{\alpha_q}^*(I^*) = \min_{I_t} \sum_{t=1}^{T} (C_{\alpha_q}^{tr}(I_t) + C_{\alpha_q}^W(I_t) + C_{\alpha_q}^E(I_t)),$$

$$s.t. \;\; C_1\colon I_t \in V, \tag{20a}$$

$$C_2\colon T^{tr}(I_t) + T^W(I_t) + T^E(I_t) \leq \phi_t^o, \tag{20b}$$

$$C_3\colon R_{V_l^{k^*}, V_{l+1}^{k^*}} \geq R_{min}, V_l^{k^*} \in V, V_{l+1}^{k^*} \in V \tag{20c}$$

where $C_{\alpha_q}^*(I^*)$ is the minimum carbon emission under the power mode α_q with the optimal reverse offloading set $I^* = \{I_1^*, \ldots, I_T^*\}$. Constraint C_1 guarantees that the offloading decision is computed in the devices of the road. Constraint C_2 guarantees that the total time for the task of the RSU is no more than its time constraint. Constraint C_3 guarantees the communication quality by letting the transmission rate of the links along the transmission path no less than R_{min}.

III. DRL-Based Vehicular Reverse Offloading Framework

In this section, we address the dynamic decision-making optimization problem by modeling it as a Markov decision process(MDP) and applying DRL methods for solutions. By integrating deep neural networks into reinforcement learning, DRL is able to extract state features from observations, enabling adaptive offloading decision-making under different power supply modes of the RSU to minimize carbon emissions. We model the MDP as follows:

1) State: We define the state to include the following information: the current queue Γ_t of tasks for all devices, the computational resources required for unit data size δ, the time constraints of the tasks ϕ_t, the carbon intensity $C_0(alpha_q)$ under different power supply modes of the RSU, the devices' position loc_i, the computational capacity f_i of the devices, the vehicles' speed v, the communication range Cov of the devices. The state is as follows:

$$s_t = \{\Gamma_t, \delta, \phi_t, C_{V_i}, C_0(\alpha_q), loc_t, f, v, Cov\} \tag{21}$$

where $\Gamma_t = \{\Gamma_i(t) \mid i = 0, \ldots, M\}$, $w_t^i = \delta D_t^i$, $loc_t = \{loc_i(t) \mid i = 0, \ldots, M\}$, $f = \{f_i \mid i = 0, \ldots, M\}$, $v = \{v_i \mid i = 1, \ldots, M\}$.

2) Action: We define the action as $a_t \in V$, which is corresponding to the offloading decision I_t.

3) Reward: We define the reward function based on the carbon emission for the task of the RSU, which is as follows:

$$r_t = \begin{cases} C_{\alpha_q}^{tr}(a_t) + C_{\alpha_q}^W(a_t) + C_{\alpha_q}^E(a_t), & \text{if } T_t^{sum} \leq \phi_t \\ \lambda, & \text{else} \end{cases} \tag{22}$$

where $T_t^{sum} = T^{tr}(a_t) + T^W(a_t) + T^E(a_t)$ and λ is the punishment value for the situation that the delay for the task is more than the time constraint ϕ_t.

Since the action is in the discrete space, we choose Deep Q-Network(DQN) [13] or Double Deep Q-Network(DDQN) [14] to obtain the optimal reverse offloading decision. DQN employs a deep neural network to approximate the Q-value function, while DDQN utilizes two networks: the main network for action selection of the current state and the target network for action evaluation of the next state-action pair.

In DQN, the Q-function is updated based on the following modified Bellman equation:

$$Q(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})) \tag{23}$$

where $Q(s_t, a_t)$ is the action-value. The network input is s_t, and the output is a vector of Q-values for all possible actions. By utilizing a deep neural network to extract the representation of the state, including the information related to carbon intensity, the decision-making process is adaptive to the power supply modes of the RSU.

TABLE I
PARAMETER SETTINGS.

Parameter	Value
v	$[30, 40]$ m/s
δ	300 cycles/bit
D	$[1, 10] \times 10^6$ MB
x, y	[0,800],[0,50] m
ϕ	$[20, 25]s$
ω_0	0.001 W/Hz
ϑ	2
f_0	[2.5, 3]×$10^4 Mcycles/s$
f_i	[2, 2.5]×$10^4 Mcycles/s$
Cov_i	200 m
B_{V_0,V_i}, B_{V_i,V_j}	$20, 40$ Mbps
$P_{V_0}^{tr}, P_{V_i}^{tr}$	$2, 0.2$ W
P_0^W, P_v^W	$1, 4.5$ W
P_0^E, P_v^E	$10, 20$ W
$C_0(\alpha_q)$	$\{11, 13, 40, 91\}$ $CO_2eq/(kWh)$
C_{V_i}	$[10, 20]$ $CO_2eq/(kWh)$
$Carbon_c$	500 CO_2eq

IV. SIMULATION RESULTS

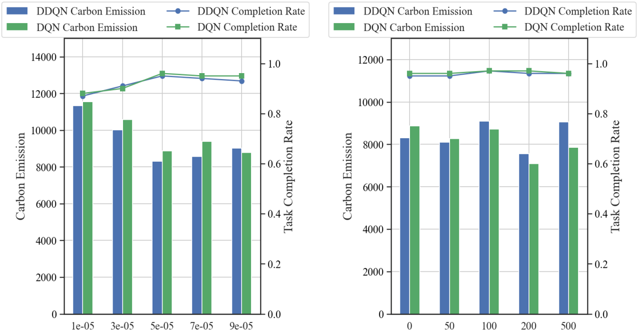
A. Simulation Settings

In this simulation, we aim to optimize carbon emissions in the vehicular reverse offloading system by deep reinforcement learning algorithms under the context of various energy supply modes. The simulation setup follows the approach described in [15], with adjustments to align with our specific application. Carbon intensity values, based on the UK's average from December 31st, 2024 to January 3rd 2025 [16], are used in the simulation, where hydropower, wind power, and solar power are associated with carbon intensities of 11, 13, 40 CO_2eqkWh, respectively, while the overall carbon intensity in UK is 91 CO_2eqkWh. These values reflect typical emissions for various power supply modes. The reinforcement learning setup employs DQN and DDQN, with training parameters including a learning rate of 0.00005, γ of 0.90, and a batch size of 64. The training utilizes a replay buffer size of 1×10^5 and updates the target network every 4 steps. Training spans 1000 episodes, and the testing covers 1000 initial states.

We compare the proposed reverse offloading method based on DQN and DDQN with the greedy and random methods, where the greedy method chooses the device with the minimum queue to offload, and the random method chooses the random device to offload.

B. Performance Evaluations

Fig. 2(a) shows the performance of DDQN and DQN in terms of carbon emissions and task completion rates under different learning rates when $M = 5$. The learning rates are set to $1 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}, 7 \times 10^{-5}, 9 \times 10^{-5}$. The results indicate that DQN exhibits similar carbon emissions with DDQN; the best learning rates for DQN and DDQN are 5×10^{-5}. As the learning rate decreases, the training fails to adequately learn, leading to deteriorating results for both DQN and DDQN. This suggests that a too-low learning rate inhibits the models from effectively exploring and optimizing, resulting in lower task completion and higher carbon emissions.



(a) Learning rate (b) Seed

Fig. 2. The carbon emissions and task completion rates under different learning rates and seeds.

Fig. 2(b) illustrates the performance under different seeds, which are set to 0, 50, 100, 200, and 500. The results show that the influence of the seed on the outcome is less significant than that of the learning rate. When the seed is set to 200, DDQN and DQN exhibit carbon emissions of 7558.2 and 7092.06, respectively, with task completion rates of 0.96 and 0.97. In contrast, the results for the greedy and random methods are 11934.85 and 14900.14 for carbon emissions, with task completion rates of 0.83 and 0.81. When comparing DQN with the greedy method, DQN's performance is notably better, with a reduction in carbon emissions by approximately 40%, and a task completion rate that is about 12% higher, which demonstrates the benefits of reinforcement learning approaches over the greedy-based method.

We set the learning rate as 5×10^{-5} and the seed as 0. In the proposed carbon emission optimization model, the power supply mode of the RSU is adaptive, while the power supply mode of the vehicle is fixed. We compared the results under several different vehicle carbon emission intensities. As shown in Fig. 3, when the carbon intensity of vehicles is 13, DQN achieves a carbon emission of 8872.76, while DDQN results in 8306.87. In contrast, Greedy and Random have emissions of 11934.85 and 14900.14, respectively. DQN reduces emissions by approximately 25.66% compared to Greedy and 40.45% compared to Random. DDQN outperforms Greedy by 30.40% and Random by 44.25%. These results demonstrate that DQN and DDQN significantly reduce carbon emissions compared to Greedy and Random at different carbon intensity levels of vehicles.

We compared the results under different number of vehicles. As shown in the Fig. 4, the results of DDQN and DQN are better than those of the Greedy and Random methods. As the number of vehicles increases, carbon emissions decrease and task completion rates improve. This is because the increase in the number of vehicles implies a larger action space and more available computing resources on the road, thus resulting in lower carbon emissions and better task completion rates.

Fig. 5 depicts the average rewards of different epochs, which illustrates the convergence of DDQN and DQN. The average

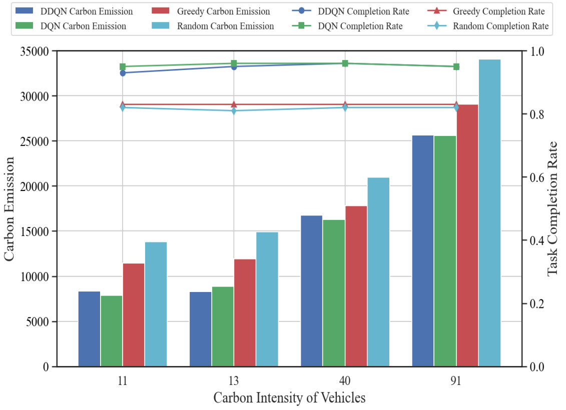


Fig. 3. The carbon emissions and task completion rates of different number of vehicles.

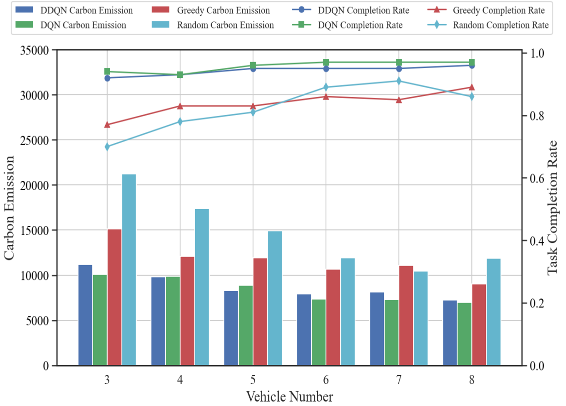


Fig. 4. The carbon emissions and task completion rates of different number of vehicles.

reward curves for both methods initially exhibit an upward trend and then stabilize as the iteration number increases.

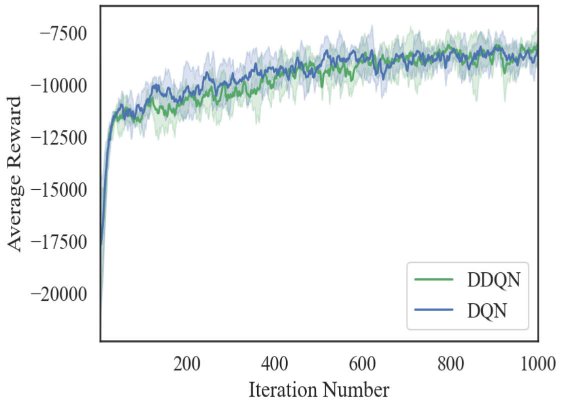


Fig. 5. The average rewards of different epochs.

V. Conclusion

In this paper, we focus on the carbon emission optimization problem in the vehicle reverse offloading system. We establish a vehicle reverse offloading system with multi-hop transmission, enabling the tasks of the RSU to be offloaded to vehicles beyond the communication range through multi-hop transmission, thereby fully utilizing the available computing resources. We further propose a power supply mode-adaptive carbon emission optimization model and optimize it by deep reinforcement learning. The experimental results demonstrate that our results have a significant advantage over the greedy-based method.

Acknowledgment

This work is supported by Engineering and Physical Sciences Research Council grant (No. EP/X040518/1 and EP/Y037421/1).

References

[1] C. Tang, G. Yan, H. Wu, and C. Zhu, “Computation offloading and resource allocation in failure-aware vehicular edge computing,” *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 1877–1888, 2024.

[2] H. Tang, H. Wu, G. Qu, and R. Li, “Double deep q-network based dynamic framing offloading in vehicular edge computing,” *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 3, pp. 1297–1310, 2023.

[3] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, “Ai-assisted network-slicing based next-generation wireless networks,” *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 45–66, 2020.

[4] Y. Liang, H. Tang, H. Wu, Y. Wang, and P. Jiao, “Lyapunov-guided offloading optimization based on soft actor-critic for isac-aided internet of vehicles,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 14 708–14 721, 2024.

[5] A. Gu, H. Wu, H. Tang, and C. Tang, “Deep reinforcement learning-guided task reverse offloading in vehicular edge computing,” in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 2200–2205.

[6] P. Amanatidis, D. Karampatzakis, G. Michailidis, T. Lagkas, and G. Iosifidis, “Adaptive reverse task offloading in edge computing for ai processes,” *Computer Networks*, vol. 255, p. 110844, 2024.

[7] Z. Song, M. Xie, J. Luo, T. Gong, and W. Chen, “A carbon-aware framework for energy-efficient data acquisition and task offloading in sustainable aiot ecosystems,” *IEEE Internet of Things Journal*, vol. 11, no. 24, pp. 39 103–39 113, 2024.

[8] Y. Yang, Y. Chen, K. Li, and J. Huang, “Carbon-aware dynamic task offloading in noma-enabled mobile edge computing for iot,” *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 15 723–15 734, 2024.

[9] W. Feng, N. Zhang, S. Li, S. Lin, R. Ning, S. Yang, and Y. Gao, “Latency minimization of reverse offloading in vehicular edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5343–5357, 2022.

[10] Y. Song, N. Zhang, and Q. J. Yet, “Deep reinforcement learning enabled reverse offloading in cooperative vehicle edge computing,” in *2024 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, 2024, pp. 862–867.

[11] Z. Yu, Y. Zhao, T. Deng, L. You, and D. Yuan, “Less carbon footprint in edge computing by joint task offloading and energy sharing,” *IEEE Networking Letters*, vol. 5, no. 4, pp. 245–249, 2023.

[12] Y.-J. Ku, S. Baidya, and S. Dey, “Adaptive computation partitioning and offloading in real-time sustainable vehicular edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 13 221–13 237, 2021.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *Computer Science*, 2013.

[14] H. v. Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI’16. AAAI Press, 2016, p. 2094–2100.

[15] W. Zhao, Y. Cheng, Z. Liu, X. Wu, and N. Kato, “Asynchronous drl based multi-hop task offloading in rsu-assisted iov networks,” *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2024.

[16] Electricity Maps, “UK 2024 Carbon Intensity Data,” https://www.electricitymaps.com/data-portal, 2024, version January 17, 2024.