Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.083294

ARTICLE

Graph-Based Constrained PPO for Low-Latency and Energy-Aware AI Agent Migration in Internet of Vehicular Agents

Kanyang Jiang[1], Yingkai Kang[2] and Ming Li[2,*]

[1]School of Automation, Guangdong University of Technology and Key Laboratory of Intelligent Detection and the Internet of Things in Manufacturing, Ministry of Education, Guangzhou, China

[2]School of Automation, Guangdong University of Technology, Guangzhou, China

*Corresponding Author: Ming Li. Email: mingli@gdut.edu.cn

Received: 01 April 2026; Accepted: 03 May 2026; Published: 15 June 2026

ABSTRACT: The Internet of Vehicular Agents (IoVA) interconnects distributed AI agents across vehicular networks to deliver real-time intelligent services for vehicular users. Due to the limited computing capacity of vehicles, AI agents are deployed on nearby RoadSide Units (RSUs) to perform computation-intensive inference. As vehicles traverse RSU coverage boundaries, AI agents must migrate to target RSUs to maintain service continuity. However, the communication and computing resources at each RSU are shared among multiple co-served vehicles, creating coupled allocation decisions that jointly determine system latency and energy consumption. To address this challenge, we propose a low-latency and energy-aware AI agent migration framework that models the end-to-end system latency and vehicle energy consumption in the IoVA. Since the cumulative nature of energy consumption introduces long-term constraints that cannot be handled by instantaneous optimization, we formulate the resource allocation problem as a constrained Markov decision process and develop a Graph-based Constrained Proximal Policy Optimization (GCPPO) algorithm to solve it. GCPPO employs a bidirectional graph attention network to extract the relational features between heterogeneous vehicles and RSUs, thereby enabling topology-aware resource allocation, and adopts a Lagrangian dual mechanism to adaptively enforce the long-term energy constraints. Simulation results demonstrate the effectiveness and scalability of the proposed algorithm, which achieves a 31.3% reduction in average system latency over baselines while attaining a 96.4% constraint satisfaction rate.

KEYWORDS: Internet of vehicular agents; AI agent migration; constrained deep reinforcement learning; graph attention network; resource allocation

1 Introduction

Large Language Models (LLMs) have demonstrated strong capabilities in natural language understanding, complex reasoning, and content generation [1,2]. AI agents leverage LLMs as their cognitive core, evolving from task-specific tools into autonomous entities capable of perceiving, reasoning, and acting across diverse domains [3]. In vehicular networks, AI agents are increasingly deployed to deliver real-time intelligent services [4,5]. However, the rapidly changing network topology and fluctuating wireless channel conditions impose stringent requirements on service continuity [6]. The Internet of Vehicular Agents (IoVA) has emerged as a promising paradigm in which distributed AI agents are seamlessly interconnected and dynamically coordinated across the vehicular environment [7]. In the IoVA, AI agents continuously perceive the surrounding environment and user requests to construct real-time situational awareness. Leveraging

this awareness, the AI agents formulate context-aware decisions and convert them into executable actions, thereby delivering intelligent services to vehicular users [7].

However, AI agent decision-making relies on computation-intensive inference that far exceeds the limited computing capacity of vehicles [8]. To sustain the inference process, AI agents are deployed on RoadSide Units (RSUs) with sufficient computing resources [9]. As vehicles continuously traverse RSU coverage boundaries, AI agents must migrate to target RSUs to maintain service continuity [10]. The resulting service latency and energy consumption are jointly determined by the allocation of communication and computing resources at the RSUs. Although increasing resource allocation can effectively reduce service latency, it also raises energy consumption, which is bounded by strict vehicle energy budgets [11]. Therefore, it remains a significant challenge to optimize resource allocation for AI agent migration in the IoVA while jointly reducing service latency and satisfying vehicle energy constraints.

Traditional optimization methods for resource allocation in vehicular networks typically rely on accurate instantaneous channel state information and quasi-static network assumptions [12,13]. Nevertheless, the high mobility of vehicles introduces significant channel estimation errors [14], and the dynamically changing network topology further limits the applicability of these methods [6,15]. Deep Reinforcement Learning (DRL)-based methods offer a promising alternative by learning effective policies without prior knowledge of system dynamics [15,16]. Despite this advantage, most existing DRL approaches encode the environment state as a concatenated observation vector [13,17]. This flat representation fails to capture the topological relationships among vehicles and RSUs, and generalizes poorly as the network scales. Furthermore, conventional DRL methods lack systematic constraint-handling mechanisms [18], making the learned policies prone to violating vehicle energy constraints in the highly dynamic IoVA.

To address the above challenges, at the system level, we develop a low-latency and energy-aware AI agent migration framework in the IoVA. The framework jointly models end-to-end service latency and vehicle energy consumption under dynamic channel conditions. We further formulate the multi-vehicle resource allocation problem as a Constrained Markov Decision Process (CMDP) with long-term cumulative energy budgets. At the algorithmic level, we design the Graph-based Constrained Proximal Policy Optimization (GCPPO) algorithm to learn an effective policy for the formulated CMDP. The main contributions of this paper are summarized as follows:

- We develop a low-latency and energy-aware AI agent migration framework in the IoVA, which jointly characterizes the end-to-end service latency and vehicle energy consumption in dynamic vehicular environments. Specifically, the latency model captures the latency across the communication, inference, and migration phases, while the energy model captures both the transmission and circuit power consumption of each vehicle over uplink and downlink channels.
- We formulate the multi-vehicle resource allocation problem for AI agent migration as a Constrained Markov Decision Process (CMDP), which aims to minimize the long-term average system latency subject to cumulative energy constraints.
- We design a novel GCPPO algorithm to solve the formulated CMDP. GCPPO leverages a bidirectional Graph Attention Network (GAT) to capture the relational features between heterogeneous vehicles and RSUs, and incorporates a Lagrangian dual method to adaptively enforce the long-term energy constraints. Simulation results demonstrate that GCPPO reduces average system latency by 31.3% compared with baselines while achieving a 96.4% constraint satisfaction rate.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 introduces the proposed low-latency and energy-aware AI agent migration framework in the IoVA. In Section 4, we

present the architecture of the GCPPO algorithm. Section 5 provides simulation results to demonstrate the performance of the GCPPO algorithm. In Section 6, we conclude the paper.

2 Related Work

2.1 AI Agents in Vehicular Networks

Driven by the rapid advancement of LLMs, researchers have increasingly explored integrating AI agents into vehicular networks [3,4,19,20]. To address the substantial computing demand of AI agents, the authors in [21] proposed a cloud-edge collaborative architecture that distributed multimodal LLM inference between edge servers and the cloud for intelligent driver assistance. Furthermore, the authors in [22] developed an Agent-as-a-Service paradigm, where AI agents autonomously performed computing and communication tasks, effectively reducing service latency for edge-assisted autonomous driving.

Given the high mobility of vehicles, AI agent migration across edge servers has emerged as a promising approach to maintain service continuity [5,23]. Specifically, the authors in [23] proposed a generative diffusion-based contract design to incentivize RSU participation in AI agent migration. To address security threats during AI agent migration, the authors in [5] developed a secure online migration framework with trust assessment, effectively mitigating network attacks while maintaining low migration latency. However, the above works focus on incentive mechanisms and migration security, while overlooking vehicle energy consumption, which critically constrains vehicles with limited resources across successive migrations in the IoVA.

2.2 Constrained Deep Reinforcement Learning for Resource Optimization

Resource optimization in wireless networks typically involves long-term constraints (e.g., energy budgets and service quality guarantees) that conventional DRL methods fail to satisfy [18,24]. Constrained DRL (CDRL) methods address this limitation by formulating the problem as a CMDP, which extends the MDP with cumulative cost constraints [25]. Among CDRL methods, Lagrangian dual optimization is the most widely adopted approach, which solves the CMDP by converting constraints into penalty terms in the objective function [18,25]. For instance, the authors in [24] employed Lagrangian dual optimization to achieve near-optimal network capacity through joint UAV altitude control and channel access under energy harvesting constraints. Since Lagrangian dual methods may still produce infeasible actions during execution [18], recent studies have incorporated safety mechanisms to provide stronger constraint guarantees [26,27]. For instance, the authors in [28] embedded a safety layer that projects each action onto the feasible set to satisfy latency constraints for edge offloading. However, the above CDRL methods rely on flat state representations, which cannot capture the spatial relationships among interacting entities. When applied to IoVA scenarios with heterogeneous vehicles and RSUs under dynamic topologies, this limitation degrades the expressiveness and scalability of the learned policies.

3 Low-Latency and Energy-Aware AI Agent Migration Framework

In this section, we present the proposed low-latency and energy-aware AI agent migration framework in the IoVA, as illustrated in Fig. 1. The framework considers a system where AI agents are deployed on RSUs to perform computation-intensive inference, and migrate to target RSUs as vehicles traverse coverage boundaries. We first model the service latency and vehicle energy consumption, and then formulate the resource allocation problem.

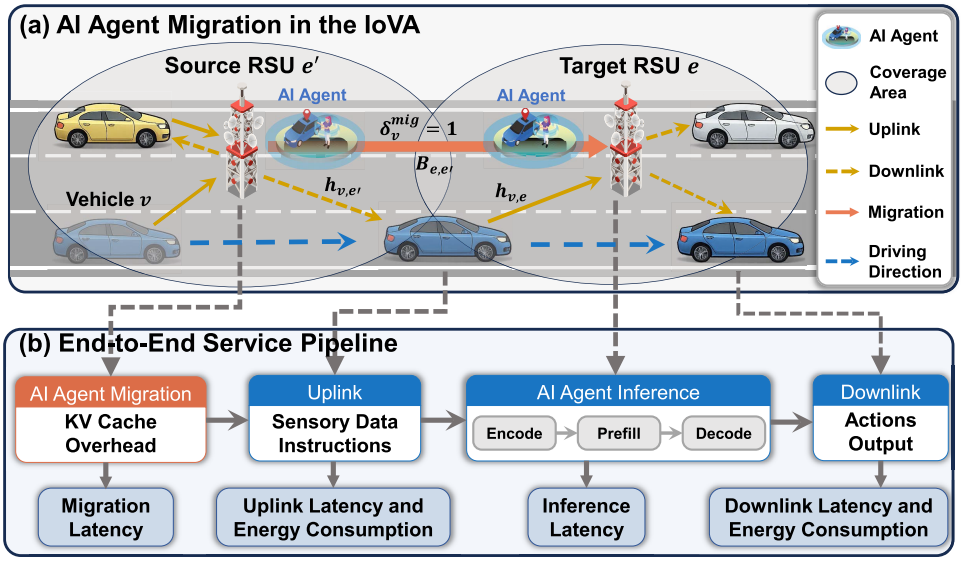

Figure 1: Illustration of the proposed low-latency and energy-aware AI agent migration framework in the IoVA.

3.1 Service Latency Model

We consider an IoVA system that consists of a set $\mathcal{V} = \{1, \ldots, v, \ldots, V\}$ of V vehicles and a set $\mathcal{E} = \{1, \ldots, e, \ldots, E\}$ of E RSUs. Due to limited on-board computing capacity, AI agents are deployed on nearby RSUs for computation-intensive inference [9]. The service duration is discretized into T time slots. Each time slot is sufficiently short that the vehicle-RSU associations and channel conditions remain approximately constant [29]. At time slot t, vehicle v is associated with its serving RSU $k_v(t) \in \mathcal{E}$, and the set of vehicles served by RSU e is denoted by $\mathcal{V}_e(t) = \{v \in \mathcal{V} \mid k_v(t) = e\}$. The service process comprises uplink transmission, RSU-side inference, AI agent migration upon RSU handover, and downlink delivery.

In the communication phases, each vehicle exchanges data with its serving RSU through shared wireless bandwidth. Let B_e^u and B_e^d denote the total uplink and downlink bandwidth of RSU e, and let $\beta_{v,e}^u(t)$ and $\beta_{v,e}^d(t)$ denote the allocation ratios for vehicle v, subject to $\sum_{v \in \mathcal{V}_e(t)} \beta_{v,e}^u(t) \le 1$ and $\sum_{v \in \mathcal{V}_e(t)} \beta_{v,e}^d(t) \le 1$. Based on the Shannon-Hartley theorem, the uplink and downlink transmission rates can be expressed as

$$R_{v,e}^{up}(t) = \beta_{v,e}^u(t) B_e^u \log_2\left(1 + \frac{p_v^u(t) h_{v,e}(t)}{\sigma_e^2}\right), \quad R_{v,e}^{dw}(t) = \beta_{v,e}^d(t) B_e^d \log_2\left(1 + \frac{p_e^d h_{v,e}(t)}{\sigma_v^2}\right), \tag{1}$$

where $p_v^u(t)$ is the uplink transmit power of vehicle v at time slot t, p_e^d is the fixed downlink transmit power of RSU e, $h_{v,e}(t)$ denotes the channel gain between vehicle v and RSU e at time slot t, and σ_e^2 and σ_v^2 represent the noise power at the RSU and the vehicle, respectively. In the uplink phase, the vehicle transmits environmental perception data of size $M_v^{sen}(t)$ and user instructions of size $M_v^{ins}(t)$. In the downlink phase, the RSU returns structured actions of size $M_v^{act}(t)$ and textual outputs of size $M_v^{txt}(t)$. The uplink and downlink communication latencies can be obtained as

$$D_{v,e}^{up}(t) = \frac{M_v^{sen}(t) + M_v^{ins}(t)}{R_{v,e}^{up}(t)}, \quad D_{v,e}^{dw}(t) = \frac{M_v^{act}(t) + M_v^{txt}(t)}{R_{v,e}^{dw}(t)}. \tag{2}$$

In the computation phase, the RSU allocates its computing resources to the AI agents for inference [8]. Specifically, the AI agent first encodes the multimodal perception data into a joint representation of the driving environment and task intent. This representation is then processed in parallel during prefill to

construct the Key-Value (KV) cache for the session. Based on this cache, the AI agent performs autoregressive decoding to generate the response. The encoding, prefill, and decode stages process data volumes of $M_v^{enc}(t)$, $M_v^{pre}(t)$, and $M_v^{dec}(t)$, respectively. Let $\alpha_{v,e}^{alc}(t)$ denote the fraction of computing resources of RSU e allocated to vehicle v, subject to $\sum_{v\in\mathcal{V}_e(t)} \alpha_{v,e}^{alc}(t) \le 1$, and let ω_e and C_e denote the computational intensity in GPU cycles per unit of data and the processing speed of RSU e, respectively. The resulting inference latency is expressed as

$$D_{v,e}^{pro}(t) = \frac{\omega_e(M_v^{enc}(t) + M_v^{pre}(t) + M_v^{dec}(t))}{\alpha_{v,e}^{alc}(t)C_e}. \tag{3}$$

Since the AI agent is deployed on the RSU, it must be migrated upon RSU handover to preserve service continuity [30]. Let $e' = k_v(t-1)$ denote the serving RSU at the previous time slot and define the migration indicator $\delta_v^{mig}(t) = \mathbb{I}[e' \neq e]$, which equals 1 if a handover occurs and 0 otherwise. As the vehicle moves, the AI agent processes newly received perception data and user instructions at each time slot, appending new entries to the KV cache at a rate of S_v^{kv} per context token. The accumulated cache of $N_v^{ctx}(t)$ context tokens is compressed via quantization with coefficient q_v to reduce the transfer volume. Together with a fixed overhead Ω_v covering the model parameters, runtime environment, and protocol data, the migration latency over the inter-RSU link with bandwidth $B_{e,e'}$ is given by

$$D_{v,e'\to e}^{mig}(t) = \frac{N_v^{ctx}(t)S_v^{kv}q_v + \Omega_v}{B_{e,e'}}. \tag{4}$$

Thus, the end-to-end service latency for vehicle v at time slot t is given by

$$D_v^{tot}(t) = D_{v,e}^{up}(t) + D_{v,e}^{pro}(t) + \delta_v^{mig}(t)D_{v,e'\to e}^{mig}(t) + D_{v,e}^{dw}(t). \tag{5}$$

3.2 Energy Consumption Model

In the IoVA, vehicles operate on limited onboard batteries while RSUs are grid-powered, we model only the vehicle-side energy consumption during uplink and downlink communication [11].

In the uplink phase, the vehicle transmits data to the serving RSU through its radio frequency chain [29]. Due to the limited amplifier efficiency, the power amplifier draws input power that exceeds the intended transmit power. The active transmit chain further consumes static circuit power from the supporting circuitry and a bandwidth-dependent dynamic component that arises from baseband signal processing across the allocated bandwidth. The resulting uplink energy consumption can be expressed as

$$E_v^{up}(t) = \left(\frac{p_v^u(t)}{\eta_v^{pa}} + p_v^{us} + \kappa^u B_{v,e}^u(t)\right) D_{v,e}^{up}(t), \tag{6}$$

where η_v^{pa} is the power amplifier efficiency, p_v^{us} is the uplink static circuit power, and κ^u is the bandwidth-dependent circuit power coefficient for the uplink chain.

In the downlink phase, the vehicle receives the service response without power amplification. The receive chain consumes a demodulation power p_v^d, a static circuit power p_v^{ds}, and a bandwidth-dependent component $\kappa^d B_{v,e}^d(t)$, yielding the downlink energy consumption

$$E_v^{dw}(t) = \left(p_v^d + p_v^{ds} + \kappa^d B_{v,e}^d(t)\right) D_{v,e}^{dw}(t). \tag{7}$$

Thus, the total energy consumption of vehicle v at time slot t is given by

$$E_v^{tot}(t) = E_v^{up}(t) + E_v^{dw}(t). \tag{8}$$

3.3 Problem Formulation

We aim to optimize the resource allocation in the IoVA to minimize the cumulative service latency across all vehicles while satisfying per-vehicle energy budget constraints. However, increasing the transmit power of a vehicle reduces its latency but raises its energy consumption, while allocating more bandwidth to one vehicle limits the resources available to others. To balance these competing objectives, we jointly determine the computing resource allocation ratio $\alpha_{v,e}^{alc}(t)$, the uplink and downlink bandwidth allocation ratios $\beta_{v,e}^{u}(t)$ and $\beta_{v,e}^{d}(t)$, and the vehicle transmit power $p_v^u(t)$, collectively denoted as $\mathcal{A} = \{\alpha_{v,e}^{alc}(t), \beta_{v,e}^{u}(t), \beta_{v,e}^{d}(t), p_v^u(t)\}_{v\in\mathcal{V},e\in\mathcal{E},t\in\mathcal{T}}$. The optimization problem can be formulated as

$$\min_{\mathcal{A}} \quad \sum_{t=1}^{T}\sum_{v=1}^{V} D_v^{tot}(t) \tag{9a}$$

$$\text{s.t.} \quad \sum_{v\in\mathcal{V}_e(t)} \alpha_{v,e}^{alc}(t) \le 1, \qquad \forall e\in\mathcal{E},\ \forall t\in\mathcal{T}, \tag{9b}$$

$$\sum_{v\in\mathcal{V}_e(t)} \beta_{v,e}^{u}(t) \le 1, \ \sum_{v\in\mathcal{V}_e(t)} \beta_{v,e}^{d}(t) \le 1, \qquad \forall e\in\mathcal{E},\ \forall t\in\mathcal{T}, \tag{9c}$$

$$p_v^u(t) \in [0, \tilde{p}_v^u], \qquad \forall v\in\mathcal{V},\ \forall t\in\mathcal{T}, \tag{9d}$$

$$k_v(t) \in \mathcal{E}_v^{com}, \qquad \forall v\in\mathcal{V},\ \forall t\in\mathcal{T}, \tag{9e}$$

$$\sum_{t=1}^{T} E_v^{tot}(t) \le E_v^{bud}, \qquad \forall v\in\mathcal{V}. \tag{9f}$$

Constraints (9b,c) ensure that the computing and bandwidth allocation ratios assigned by each RSU do not exceed unity. Constraint (9d) bounds the uplink transmit power of each vehicle in its feasible range, where $\tilde{p}_v^u$ denotes the maximum transmit power. Constraint (9e) restricts the serving RSU of each vehicle to its communicable RSU set $\mathcal{E}_v^{com}$. Constraint (9f) imposes that the cumulative energy consumption of each vehicle over the entire service period does not exceed its energy budget E_v^{bud}.

4 Graph-Based Constrained Proximal Policy Optimization Algorithm

4.1 CMDP Formulation

The resource allocation problem in the IoVA involves a non-convex objective, temporally coupled energy constraints, and non-stationary system dynamics, which render conventional approaches intractable. We therefore reformulate it as a CMDP [25], characterized by the tuple $\langle \mathcal{S}, \mathcal{A}, r, c, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, r is the reward function, c is the cost function, and $\gamma \in (0,1]$ is the discount factor. The detailed definitions are as follows:

(1) *State space:* At each time slot t, the CDRL agent observes the system state $\mathbf{s}(t) = \{h_{v,e}(t), k_v(t), \mathbf{m}_v(t), N_v^{ctx}(t), \hat{E}_v(t)\}_{v\in\mathcal{V},e\in\mathcal{E}}$, which encompasses channel gains, RSU associations, service data volumes, KV cache sizes, and cumulative energy consumption. Here, $\mathbf{m}_v(t)$ collects all service data volumes of vehicle v defined in Section 3.1, and $\hat{E}_v(t) = \sum_{\tau=1}^{t-1} E_v^{tot}(\tau)$ tracks the energy consumed up to time slot t.

(2) *Action space:* At each time slot t, the CDRL agent determines the action $\mathbf{a}(t) = \{\alpha^{alc}_{v,e}(t), \beta^{u}_{v,e}(t), \beta^{d}_{v,e}(t), p^{u}_{v}(t)\}_{v\in\mathcal{V}}$, which jointly specifies the computing, bandwidth, and power allocation for each vehicle.

(3) *Reward function:* Since the goal is to minimize the cumulative service latency, the immediate reward is defined as $r(t) = -\sum_{v=1}^{V} D^{tot}_{v}(t)$.

(4) *Cost function:* Among the constraints in (9), constraints (9b–d) are enforced via per-slot action projection, and constraint (9e) is determined by the physical distance between each vehicle and its nearest RSU. However, constraint (9f) couples decisions across all time slots. To encode this long-term constraint, we define the immediate cost as $c_v(t) = E^{tot}_{v}(t)$. Accordingly, the CMDP optimization objective is to maximize the expected cumulative reward subject to the energy budget constraint (9f), which can be formulated as

$$\max_{\pi} \quad J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} \gamma^{t-1} r(t)\right] \tag{10a}$$

$$\text{s.t.} \quad J^{v}_{c}(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} \gamma^{t-1} c_v(t)\right] \le E^{bud}_{v}, \qquad \forall v \in \mathcal{V}. \tag{10b}$$

4.2 Architecture of the GCPPO Algorithm

4.2.1 Bidirectional Graph Attention Network for State Representation

To capture the relational structure among vehicles and RSUs in the IoVA, we represent the system state as a bipartite graph $\mathcal{G}(\mathbf{s}) = (\mathcal{V} \cup \mathcal{E}, \mathcal{L})$. Each vehicle node $v \in \mathcal{V}$ carries a feature vector $\mathbf{x}_v$ that encodes its channel condition, service demands, and energy status, and each RSU node $e \in \mathcal{E}$ carries $\mathbf{x}_e$ that encodes its computing and communication capacity. The edge set $\mathcal{L} = \{(v, e) \mid v \in \mathcal{V}_e(t)\}$ connects each vehicle to its serving RSU, which encodes the resource-sharing structure [17].

The bipartite graph $\mathcal{G}$ is then processed by a bidirectional GAT [31]. The node features serve as the initial embeddings, i.e., $\mathbf{h}^{(0)}_{v} = \mathbf{x}_v$ and $\mathbf{h}^{(0)}_{e} = \mathbf{x}_e$. At layer l, the bidirectional aggregations are given by

$$\mathbf{h}^{(l+1)}_{v} = \sigma\left(\sum_{e\in\mathcal{N}_v} \alpha^{(l)}_{v,e} \mathbf{W}^{(l)}_{e} \mathbf{h}^{(l)}_{e}\right), \quad \mathbf{h}^{(l+1)}_{e} = \sigma\left(\sum_{v\in\mathcal{N}_e} \alpha^{(l)}_{e,v} \mathbf{W}^{(l)}_{v} \mathbf{h}^{(l)}_{v}\right), \tag{11}$$

where $\mathcal{N}_v$ is the set of RSU nodes adjacent to vehicle v, $\mathcal{N}_e = \mathcal{V}_e(t)$ is the set of vehicles served by RSU e, $\mathbf{W}^{(l)}_{e}$ and $\mathbf{W}^{(l)}_{v}$ are direction-specific learnable weight matrices, and $\sigma(\cdot)$ denotes the activation function. The attention coefficient $\alpha^{(l)}_{i,j}$ is computed as

$$\alpha^{(l)}_{i,j} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^{(l)\top}[\mathbf{W}^{(l)}\mathbf{h}^{(l)}_{i} \| \mathbf{W}^{(l)}\mathbf{h}^{(l)}_{j}]\right)\right)}{\sum_{j'\in\mathcal{N}_i} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^{(l)\top}[\mathbf{W}^{(l)}\mathbf{h}^{(l)}_{i} \| \mathbf{W}^{(l)}\mathbf{h}^{(l)}_{j'}]\right)\right)}, \tag{12}$$

where i denotes the target node, $j \in \mathcal{N}_i$ denotes its neighboring nodes, $\mathbf{a}^{(l)}$ is a learnable attention vector, and the parameters $(\mathbf{W}^{(l)}, \mathbf{a}^{(l)})$ take direction-specific values $(\mathbf{W}^{(l)}_{e}, \mathbf{a}^{(l)}_{e})$ for RSU-to-vehicle edges and $(\mathbf{W}^{(l)}_{v}, \mathbf{a}^{(l)}_{v})$ for vehicle-to-RSU edges. The resulting vehicle embeddings thus reflect the resource availability of nearby RSUs, while the RSU embeddings capture the aggregate demand of served vehicles.

Since the aggregations in (11) only access immediate neighbors, we stack L layers to expand the receptive field to L-hop neighbors, which enables each vehicle node to incorporate information from co-served vehicles that compete for the same RSU resources. After the final layer, we concatenate all vehicle node

embeddings to obtain the state vector $\mathbf{z} = [\mathbf{h}_v^{(L)}]_{v\in\mathcal{V}}$, which serves as the input to the actor and critic networks of the CDRL agent.

4.2.2 Graph-Based PPO for Energy-Constrained AI Agent Migration

The actor network $\pi_\theta(\mathbf{a} \mid \mathbf{z})$ parameterizes a Gaussian policy over the continuous action space, where separate fully connected layers output the mean $\boldsymbol{\mu}_\theta(\mathbf{z})$ and standard deviation $\boldsymbol{\sigma}_\theta(\mathbf{z})$. At time slot t, the actor outputs a raw action $\tilde{\mathbf{a}}(t)$, which is then projected onto the feasible set defined by (9b–d) prior to execution. Specifically, since constraints (9b,c) couple the per-RSU shares across co-served vehicles, we apply a per-RSU softmax over $\mathcal{V}_e(t)$ that maps $\tilde{\alpha}_{v,e}$ onto the simplex via $\alpha_{v,e}^{alc}(t) = \exp(\tilde{\alpha}_{v,e})/\sum_{v'\in\mathcal{V}_e(t)}\exp(\tilde{\alpha}_{v',e})$. The bandwidth shares $\beta_{v,e}^u(t)$ and $\beta_{v,e}^d(t)$ are obtained in the same way, while the per-vehicle transmit power is mapped via $p_v^u(t) = \tilde{p}_v^u\,\mathrm{sigmoid}(\tilde{p}_v)$ to satisfy (9d). The resulting projection is a deterministic mapping applied to the sampled action, thereby preserving the validity of the PPO policy gradient computed on $\tilde{\mathbf{a}}(t)$ and ensuring strict feasibility. To support the policy update, the cumulative reward and energy cost under the current policy are evaluated by a reward critic $V_\phi^r(\mathbf{z})$ and a cost critic $V_\psi^{c,v}(\mathbf{z})$, respectively. Based on the cost critic estimates, we adopt Lagrangian relaxation [25] to enforce the long-term energy budget constraint in (10b), introducing a non-negative Lagrange multiplier λ_v for each vehicle v to penalize energy budget violations. The Lagrangian objective and the corresponding per-step advantage can be expressed as

$$\mathcal{L}(\theta,\boldsymbol{\lambda}) = J(\pi_\theta) - \sum_{v=1}^{V}\lambda_v\left(J_c^v(\pi_\theta) - E_v^{bud}\right), \quad \hat{A}^L(t) = \hat{A}^r(t) - \sum_{v=1}^{V}\lambda_v\hat{A}^{c,v}(t), \tag{13}$$

where $\boldsymbol{\lambda} = [\lambda_1,\ldots,\lambda_V]^\top$ denotes the vector of Lagrange multipliers, and $\hat{A}^r(t)$ and $\hat{A}^{c,v}(t)$ are the reward and cost advantages estimated via Generalized Advantage Estimation (GAE) [32]. As λ_v increases, the policy is steered away from energy-intensive actions toward more conservative resource allocation. To update the policy, we adopt the PPO clipped surrogate objective [33] with clipping parameter ε and importance sampling ratio $\rho(t) = \pi_\theta(\mathbf{a}(t) \mid \mathbf{z}(t))/\pi_{\theta_{old}}(\mathbf{a}(t) \mid \mathbf{z}(t))$. The GCPPO policy objective can be expressed as

$$L^{GCPPO}(\theta) = \hat{\mathbb{E}}_t\left[\min\left(\rho(t)\hat{A}^L(t),\ \mathrm{clip}(\rho(t),1-\epsilon,1+\epsilon)\hat{A}^L(t)\right)\right]. \tag{14}$$

Since both the policy objective and the constraint enforcement rely on accurate value estimates, the reward critic and cost critic are trained to minimize the mean squared error between their predictions and the empirical returns, with the critic loss functions given by

$$L^{V_r}(\phi) = \hat{\mathbb{E}}_t\left[\left(V_\phi^r(\mathbf{z}(t)) - \hat{R}(t)\right)^2\right], \quad L^{V_c}(\psi) = \hat{\mathbb{E}}_t\left[\sum_{v=1}^{V}\left(V_\psi^{c,v}(\mathbf{z}(t)) - \hat{C}_v(t)\right)^2\right], \tag{15}$$

where $\hat{R}(t)$ and $\hat{C}_v(t)$ denote the discounted cumulative reward and cost returns computed from the collected trajectories, respectively. After each policy update, the Lagrange multipliers are updated via dual gradient ascent based on the cost critic estimates as

$$\lambda_v \leftarrow \max\left(0,\ \lambda_v + \eta_\lambda\left(\hat{J}_c^v - E_v^{bud}\right)\right), \tag{16}$$

where η_λ is the dual learning rate and $\hat{J}_c^v$ is the estimated cumulative energy cost of vehicle v under the current policy. The overall training procedure is presented in Algorithm 1, with per-iteration training complexity $O(TL(V+E)d^2 + KTVd\cdot\max(d_a,d_c))$ [31] and per-step inference complexity $O(L(V+E)d^2 + Vd\cdot d_a)$, where d is the GAT embedding dimension, d_a and d_c are the hidden dimensions of the actor and critic networks, and K is the number of policy update epochs.

Algorithm 1: GCPPO training procedure

Initialize actor parameters θ, reward critic parameters ϕ, cost critic parameters ψ, and Lagrange multipliers $\lambda_v \leftarrow 0,\ \forall v \in \mathcal{V}$
for episode = 1, 2, . . . **do**
  Reset environment and observe initial state **s**
  for $t = 1, 2, \ldots, T$ **do**
    Construct bipartite graph $\mathcal{G}(\mathbf{s}(t))$ and compute $\mathbf{z}(t)$ via bidirectional GAT
    Sample action $\mathbf{a}(t) \sim \pi_\theta(\cdot \mid \mathbf{z}(t))$
    Project $\mathbf{a}(t)$ onto the feasible set defined by (9b–d)
    Execute $\mathbf{a}(t)$, observe reward *r(t)*, costs $\{c_v(t)\}_{v\in\mathcal{V}}$, and next state $\mathbf{s}(t+1)$
  end for
  Compute reward advantages $\hat{A}^r(t)$ and cost advantages $\hat{A}^{c,v}(t)$ using GAE
  Compute Lagrangian advantages $\hat{A}^L(t)$ via (13)
  for $k = 1, 2, \ldots, K$ **do**
    Update actor θ by maximizing $L^{GCPPO}(\theta)$ in (14)
    Update reward critic ϕ and cost critic ψ by minimizing (15)
  end for
  Update λ_v for all $v \in \mathcal{V}$ via (16)
end for
return Trained policy π_{θ^*}

5 Simulation Results

5.1 Experimental Setup

We consider an IoVA system in which vehicles travel along a road segment at speeds of 30 to 120 km/h and are served by uniformly deployed RSUs. The initial positions of vehicles are randomly distributed along the road, and each vehicle maintains a constant speed throughout each episode. Channel gains between each vehicle and RSU combine a distance-dependent path loss with exponent $\alpha = 3.76$ and Rayleigh small-scale fading [14]. The uplink sensing and instruction data volumes range over $[1, 15]$ and $[0.02, 0.625]$ MB, respectively, while the downlink actuation and text response volumes range over $[0.001, 0.05]$ and $[0.004, 0.2]$ MB, respectively. The inference data volumes range over $[0.1, 6.25]$ MB across the encoding, prefill, and decoding stages, and the context token counts range over $[1024, 8192]$. The detailed simulation parameters are summarized in Table 1.

Table 1: Simulation parameters.

Parameter	Value	Parameter	Value
Environment settings		*GCPPO hyperparameters*	
Number of vehicles	$[20, 100]$	Number of GAT layers	2
Number of RSUs	6	GAT embedding dimension	128
Uplink and downlink bandwidth	$[20, 120]$ MHz	Hidden dimension of actor and critic	128
Max vehicle transmit power	$[0.1, 0.5]$ W	Actor learning rate	2.4×10^{-4}
RSU downlink transmit power	1.8 W	Critic learning rate	3×10^{-4}
Receiver noise power	−100 dBm	Discount factor	0.99
RSU processing speed	7.8×10^{10} cycles/s	PPO clipping parameter	0.2
Inter-RSU link bandwidth	500 MHz	Policy update epochs per episode	3
Energy constraint threshold	$\{160, 200, 240\}$ J	Lagrange multiplier learning rate	0.035

For the GCPPO algorithm, the actor and critic networks share a two-layer bidirectional GAT encoder. All algorithms are implemented in PyTorch and trained for 5×10^5 environment steps on an NVIDIA RTX 4090 GPU with 5 random seeds. Each algorithm's final policy is evaluated on 20 test episodes per seed, and the mean and standard deviation of each performance metric are computed over the resulting 100 episodes. The GCPPO algorithm completes training in 17.87 min. During inference, it takes 1.44 ms on average to generate a joint resource allocation decision.

To validate the effectiveness of the proposed GCPPO algorithm in the IoVA, we compare it with the following baseline algorithms:

- **CPO [34]:** Constrained Policy Optimization (CPO) solves the CMDP via trust region updates with linearized cost constraints, providing first-order feasibility guarantees.
- **PPO-Lag [35]:** PPO-Lagrangian (PPO-Lag) augments the standard PPO objective with a Lagrangian penalty for energy constraint enforcement. It serves as an ablation variant of GCPPO without the bidirectional GAT encoder, isolating the contribution of the graph-based state representation.
- **SAC-Lag [35,36]:** Soft Actor-Critic with Lagrangian (SAC-Lag) extends the entropy-regularized off-policy framework with dual variables for cost constraint satisfaction.
- **DAPA:** Demand-Aware Proportional Allocation (DAPA) is a heuristic baseline that allocates computing and bandwidth resources at each RSU in proportion to the service demand of served vehicles, and adjusts each vehicle's transmit power according to its demand priority and residual energy.
- **Random:** Random uniformly samples resource allocation decisions from the feasible action space.

5.2 Performance Evaluation

In Fig. 2, we present the training performance of GCPPO and the baseline algorithms. As shown in Fig. 2a, GCPPO achieves the highest converged reward and converges in 1×10^5 training steps, substantially faster than all baselines. Among the constrained baselines, CPO and PPO-Lag plateau at lower rewards, while SAC-Lag exhibits high variance due to the instability of off-policy Lagrangian updates. Fig. 2b illustrates the cost curves under an energy budget of 200 J. GCPPO maintains its cost consistently below the energy constraint after initial training, whereas CPO converges below the threshold but adopts a more conservative policy that limits reward improvement. In contrast, SAC-Lag frequently exceeds the budget. Fig. 2c depicts the convergence trajectories in the reward-cost plane, where GCPPO converges to the upper-left corner of the feasible region, achieving the highest reward while satisfying the energy budget. The superior performance of GCPPO is attributed to the combined effect of the bidirectional GAT, which improves reward through topology-aware allocation, and the Lagrangian dual mechanism, which constrains the trajectory in the feasible region.

In Fig. 3, we present the evaluation results on independent test episodes. As shown in Fig. 3a, GCPPO achieves the lowest average system latency, reducing it by 36.0% over SAC-Lag and by approximately 4% over CPO and PPO-Lag. Fig. 3b shows that GCPPO matches CPO in energy consumption while consuming 20.9% less than SAC-Lag. As illustrated in Fig. 3c, GCPPO achieves a constraint satisfaction rate of 96.4%, substantially outperforming PPO-Lag and SAC-Lag. Although DAPA attains the highest satisfaction rate of 99.2%, this comes at a 76.6% increase in system latency. These results indicate that the bidirectional GAT encoder enables GCPPO to exploit the vehicle-RSU topology for effective resource allocation, while the Lagrangian dual mechanism maintains energy feasibility without sacrificing latency performance.

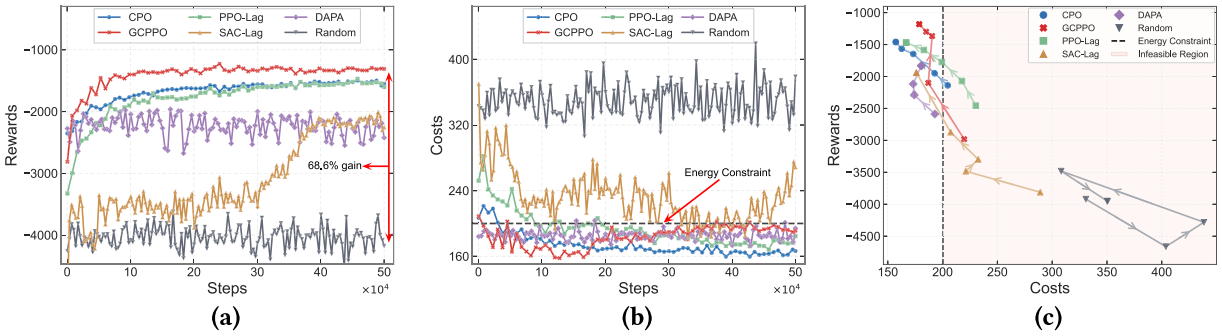

Figure 2: Comparison of training performance among GCPPO and baseline algorithms. (**a**) Reward curves for different algorithms. (**b**) Cost curves for different algorithms under the energy constraint threshold. (**c**) Convergence trajectories of different algorithms in the reward-cost plane.

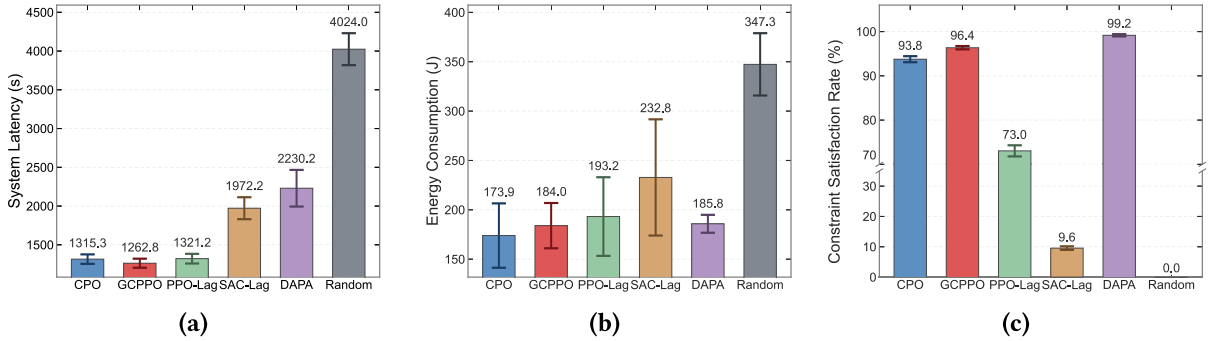

Figure 3: Performance evaluation of GCPPO and baseline algorithms for low-latency and energy-aware AI agent migration optimization. (**a**) Average system latency. (**b**) Average energy consumption. (**c**) Constraint satisfaction rate.

5.3 Sensitivity Analysis

Fig. 4 illustrates the impact of the energy budget on GCPPO in the IoVA. As the budget increases from 160 to 240 J, the converged reward improves from approximately −1500 to −1200, and the constraint satisfaction rate rises from 90.8% to 98.1%. This is because a more relaxed budget allows the CDRL agent to allocate higher transmit power for latency reduction without exceeding the energy limit. Under the tightest budget of 160 J, the Lagrangian multipliers automatically increase, yielding a more energy-conservative policy that trades higher latency for constraint satisfaction.

Figs. 5 and 6 illustrate the impact of wireless bandwidth and vehicle transmit power in the IoVA, respectively. As bandwidth increases from 20 to 120 MHz, both system latency and energy consumption decrease across all algorithms, since higher bandwidth improves the transmission rate, thereby reducing communication time and the associated energy expenditure. Conversely, increasing the transmit power from 0.1 to 0.5 W reduces system latency but simultaneously raises energy consumption. Under both parameter sweeps, GCPPO consistently achieves the lowest normalized average latency, outperforming the strongest baseline by approximately 12% in the bandwidth sweep. This advantage grows under resource-scarce conditions, where topology-aware allocation via the bidirectional GAT yields the largest gains. In contrast, SAC-Lag consumes approximately 80% more normalized energy than GCPPO in the power sweep, indicating that its off-policy Lagrangian enforcement cannot effectively regulate the growing energy cost.

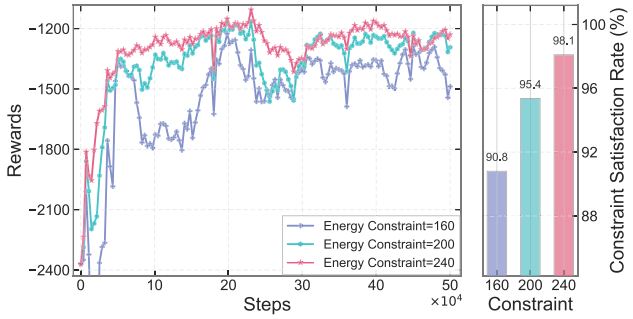

Figure 4: Reward curves and constraint satisfaction rates of the GCPPO algorithm under different energy budgets.

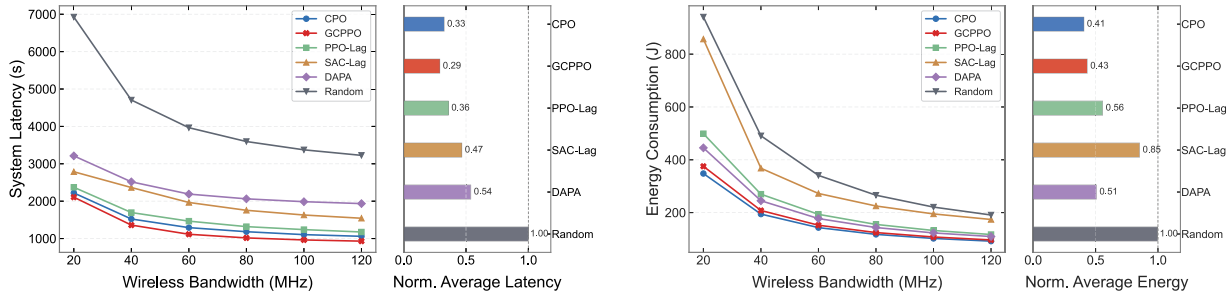

Figure 5: Impact of wireless bandwidth on the performance of GCPPO and baseline algorithms. (**a**) System latency curves and normalized average latency. (**b**) Energy consumption curves and normalized average energy consumption.

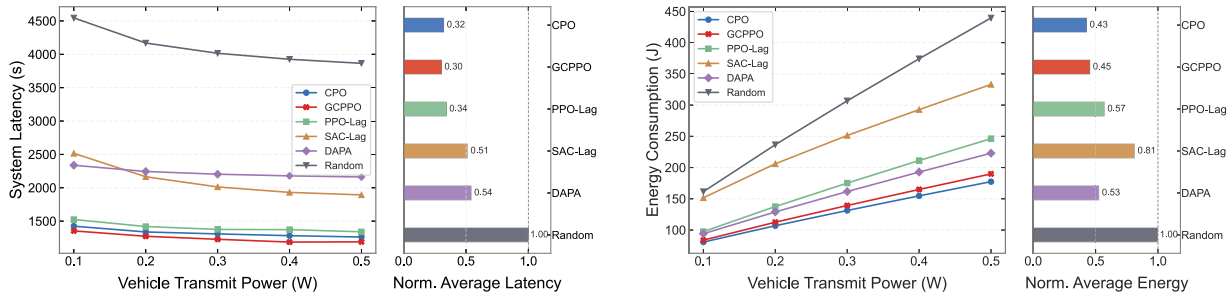

Figure 6: Impact of vehicle transmit power on the performance of GCPPO and baseline algorithms. (**a**) System latency curves and normalized average latency. (**b**) Energy consumption curves and normalized average energy consumption.

5.4 Scalability Analysis

In Fig. 7, we evaluate the scalability of all algorithms as the number of vehicles increases from 20 to 100. We quantify it using the scalability exponent α, estimated via least-squares linear regression in log-log space, where per-vehicle metrics scale as $O(V^{\alpha})$ with the number of vehicles V. An exponent below 1.0 indicates sub-linear growth, meaning that per-vehicle metrics grow more slowly than the network size.

As shown in Fig. 7a, GCPPO achieves the lowest latency scalability exponent of 0.91, indicating sub-linear growth in per-vehicle latency as the network expands. CPO, PPO-Lag, and SAC-Lag also achieve sub-linear exponents, while DAPA and Random exhibit super-linear growth. For energy consumption, Fig. 7b shows that GCPPO attains the lowest exponent of 0.60, significantly outperforming SAC-Lag

and Random, both of which approach or exceed linear growth. The sub-linear scaling of GCPPO is attributed to the bidirectional GAT encoder, which generalizes allocation decisions to larger vehicle populations, and the Lagrangian multiplier updates, which adaptively tighten the energy penalty as network density increases.

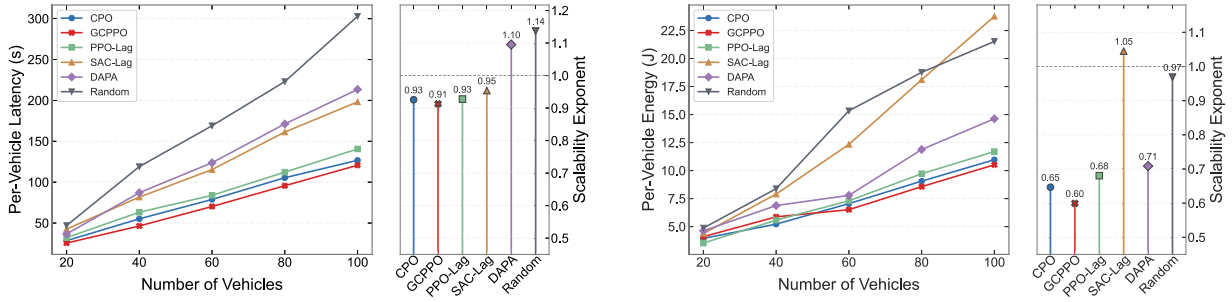

Figure 7: Impact of the number of vehicles on the performance of GCPPO and baseline algorithms, where the scalability exponent is estimated via least-squares linear regression in log-log space. (**a**) Per-vehicle latency curves and scalability exponent. (**b**) Per-vehicle energy consumption curves and scalability exponent.

6 Conclusion

In this paper, we have proposed a low-latency and energy-aware AI agent migration framework in the IoVA, jointly characterizing the end-to-end service latency and vehicle energy consumption across communication, inference, and migration phases. To solve the formulated CMDP, we have designed the GCPPO algorithm, which leverages a bidirectional GAT encoder to capture the relational structure among vehicles and RSUs, thereby enabling topology-aware resource allocation. It further incorporates a Lagrangian dual mechanism to adaptively enforce the long-term energy constraints without requiring predefined penalty weights. Simulation results demonstrate the effectiveness and scalability of GCPPO, which achieves a 31.3% reduction in average system latency over baselines while attaining a 96.4% constraint satisfaction rate. In future work, we plan to extend the proposed framework to multi-agent scenarios where distributed CDRL agents collaboratively optimize AI agent migration across heterogeneous edge networks.

Acknowledgement: None.

Funding Statement: This work was supported by the 2024 Guangdong Province Education Science Planning Project (Higher Education Special Project) under Grant 2024GXJK621.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Kanyang Jiang, Yingkai Kang and Ming Li; methodology, Kanyang Jiang and Yingkai Kang; software, Kanyang Jiang and Yingkai Kang; validation, Kanyang Jiang, Yingkai Kang and Ming Li; formal analysis, Kanyang Jiang and Yingkai Kang; investigation, Kanyang Jiang and Yingkai Kang; resources, Ming Li; data curation, Kanyang Jiang and Yingkai Kang; writing—original draft preparation, Kanyang Jiang and Yingkai Kang; writing—review and editing, Ming Li; visualization, Kanyang Jiang and Yingkai Kang; supervision, Ming Li; project administration, Ming Li. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chang Y, Wang X, Wang J, Wu Y, Yang L, Zhu K, et al. A survey on evaluation of large language models. ACM Trans Intell Syst Technol. 2024;15(3):39. doi:10.1145/3641289.
2. Plaat A, Wong A, Verberne S, Broekens J, Van Stein N, Bäck T. Multi-step reasoning with large language models, a survey. ACM Comput Surv. 2025;58(6):160. doi:10.1145/3774896.
3. Guo T, Chen X, Wang Y, Chang R, Pei S, Chawla NV, et al. Large language model based multi-agents: a survey of progress and challenges. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24. CA, USA: International Joint Conferences on Artificial Intelligence Organization; 2024. p. 8048–57.
4. Mahmud D, Hajmohamed H, Almentheri S, Alqaydi S, Aldhaheri L, Khalil RA, et al. Integrating LLMs with ITS: recent advances, potentials, challenges, and future directions. IEEE Trans Intell Transp Syst. 2025;26(5):5674–709. doi:10.1109/TITS.2025.3528116.
5. Wen X, Wen J, Xiao M, Kang J, Zhang T, Li X, et al. Defending against network attacks for secure AI agent migration in vehicular metaverses. IEEE Internet Things J. 2026;13(3):4153–66. doi:10.1109/JIOT.2025.3633501.
6. Clancy J, Mullins D, Deegan B, Horgan J, Ward E, Eising C, et al. Wireless access for V2X communications: research, challenges and opportunities. IEEE Commun Surv Tutor. 2024;26(3):2082–119. doi:10.1109/COMST.2024.3384132.
7. Wang Y, Guo S, Pan Y, Su Z, Chen F, Luan TH, et al. Internet of agents: fundamentals, applications, and challenges. IEEE Trans Cognit Commun Netw. 2026;12:4476–501. doi:10.1109/TCCN.2025.3623369.
8. Zheng Y, Chen Y, Qian B, Shi X, Shu Y, Chen J. A review on edge large language models: design, execution, and applications. ACM Comput Surv. 2025;57(8):1–35. doi:10.1145/3719664.
9. Qu G, Chen Q, Wei W, Lin Z, Chen X, Huang K. Mobile edge intelligence for large language models: a contemporary survey. IEEE Commun Surv Tutor. 2025;27(6):3820–60. doi:10.36227/techrxiv.172115025.57884352/v1.
10. Chen Z, Huang S, Min G, Ning Z, Li J, Zhang Y. Mobility-aware seamless service migration and resource allocation in multi-edge IoV systems. IEEE Trans Mob Comput. 2025;24(7):6315–32. doi:10.1109/TMC.2025.3540407.
11. Qiu B, Wang Y, Xiao H, Zhang Z. Deep reinforcement learning-based adaptive computation offloading and power allocation in vehicular edge computing networks. IEEE Trans Intell Transp Syst. 2024;25(10):13339–49. doi:10.1109/TITS.2024.3391831.
12. Shui T, Saad W, Hu Y, Chen M. Resilient vehicular communications under imperfect channel state information. IEEE Trans Wirel Commun. 2026;25:6442–59. doi:10.1109/twc.2025.3625199.
13. Xu Y, Zhu K, Xu H, Ji J. Deep reinforcement learning for multi-objective resource allocation in multi-platoon cooperative vehicular networks. IEEE Trans Wirel Commun. 2023;22(9):6185–98. doi:10.1109/twc.2023.3240425.
14. Wang P, Wu W, Liu J, Chai G, Feng L. Joint spectrum and power allocation for V2X communications with imperfect CSI. IEEE Trans Vehic Technol. 2023;72(12):16338–53. doi:10.1109/tvt.2023.3299691.
15. Ju Y, Chen Y, Cao Z, Liu L, Pei Q, Xiao M, et al. Joint secure offloading and resource allocation for vehicular edge computing network: a multi-agent deep reinforcement learning approach. IEEE Trans Intell Transp Syst. 2023;24(5):5555–69. doi:10.1109/TITS.2023.3242997.
16. Li P, Wang X, Li C, Iqbal M, Al-Dulaimi A, Chih-Lin I, et al. Deep reinforcement learning-based task scheduling and resource allocation for vehicular edge computing: a survey. IEEE Tran Intell Transp Syst. 2025;26(12):21472–501. doi:10.1109/tits.2025.3607910.
17. Ji M, Wu Q, Fan P, Cheng N, Chen W, Wang J, et al. Graph neural networks and deep reinforcement learning-based resource allocation for V2X communications. IEEE Internet Things J. 2025;12(4):3613–28. doi:10.1109/JIOT.2024.3469547.
18. Wachi A, Shen X, Sui Y. A survey of constraint formulations in safe reinforcement learning. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24. CA, USA: International Joint Conferences on Artificial Intelligence Organization; 2024. p. 8262–71.
19. Zhang R, Xiong K, Du H, Niyato D, Kang J, Shen X, et al. Generative AI-enabled vehicular networks: fundamentals, framework, and case study. IEEE Netw. 2024;38(4):259–67. doi:10.1109/MNET.2024.3391767.

20. Xie G, Xiong Z, Zhang X, Xie R, Guo S, Guizani M, et al. GAI-IoV: bridging generative AI and vehicular networks for ubiquitous edge intelligence. IEEE Trans Wirel Commun. 2024;23(10):12799–814. doi:10.1109/TWC.2024.3396276.
21. Hu Y, Ye D, Kang J, Wu M, Yu R. A cloud-edge collaborative architecture for multimodal LLM-based advanced driver assistance systems in IoT networks. IEEE Internet Things J. 2025;12(10):13208–21. doi:10.1109/jiot.2024.3509628.
22. Li B, Liu T, Wang W, Zhao C, Wang S. Agent-as-a-service: an AI-native edge computing framework for 6G networks. IEEE Netw. 2025;39(2):44–51. doi:10.1109/mnet.2024.3520987.
23. Zhong Y, Kang J, Wen J, Ye D, Nie J, Niyato D, et al. Generative diffusion-based contract design for efficient AI twin migration in vehicular embodied AI networks. IEEE Trans Mobile Comput. 2025;24(5):4573–88. doi:10.1109/tmc.2025.3526230/mm1.
24. Khairy S, Balaprakash P, Cai LX, Cheng Y. Constrained deep reinforcement learning for energy sustainable multi-UAV based random access IoT networks with NOMA. IEEE J Selected Areas Commun. 2021;39(4):1101–15. doi:10.1109/JSAC.2020.3018804.
25. Altman E. Constrained markov decision processes. Abingdon, UK: Routledge; 1999. doi:10.1201/9781315140223.
26. Koursioumpas N, Magoula L, Petropouleas N, Thanopoulos AI, Panagea T, Alonistioti N, et al. A safe deep reinforcement learning approach for energy efficient federated learning in wireless communication networks. IEEE Trans Green Commun Netw. 2024;8(4):1862–74. doi:10.1109/TGCN.2024.3372695.
27. Gao Z, Hao H, Gao F, Zhao R. Constrained reinforcement-learning-enabled policies with augmented lagrangian for cooperative intersection management. IEEE Internet Things J. 2025;12(5):5396–411. doi:10.1109/jiot.2024.3487854.
28. Huang H, Ye Q, Zhou Y. Safety-critical offloading with constrained reinforcement learning for multi-access edge computing. ACM Trans Sens Netw. 2025;21(2):1–37. doi:10.1145/3715695.
29. Jang Y, Jeong S, Kang J. Energy-efficient vehicular edge computing with one-by-one access scheme. IEEE Wirel Commun Lett. 2024;13(1):39–43. doi:10.1109/LWC.2023.3318632.
30. Kang Y, Wen J, Kang J, Zhang T, Du H, Niyato D, et al. Hybrid-generative diffusion models for attack-oriented twin migration in vehicular metaverses. IEEE Trans Vehic Technol. 2025;74(9):14720–34. doi:10.1109/tvt.2025.3566034.
31. Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. arXiv:1710.10903. 2018.
32. Schulman J, Moritz P, Levine S, Jordan MI, Abbeel P. High-dimensional continuous control using generalized advantage estimation. arXiv:1506.02438. 2016.
33. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv:1707.06347. 2017.
34. Achiam J, Held D, Tamar A, Abbeel P. Constrained policy optimization. In: Proceedings of the 34th International Conference on Machine Learning. London, UK: PMLR; 2017. p. 22–31.
35. Ray A, Achiam J, Amodei D. Benchmarking safe exploration in deep reinforcement learning. arXiv:1910.01708. 2019.
36. Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proceedings of the 35th International Conference on Machine Learning. London, UK: PMLR; 2018. p. 1861–70.