Distributed Task Offloading Using Federated Reinforcement Learning in VEC

Namory Fofana[1‡], Wiem Fekih Hassen[2δ], Asma Ben Letaifa[1‡] and Abderrezak Rachedi[3⋆]
[1]Mediatron Lab (LR11TIC05), Sup'Com Tunis, University of Carthage, 2083 Ariana, Tunisia
[2]Chair of Distributed Information System, University of Passau, 94032 Passau, Germany
[3]Computer Science Lab (LIGM, CNRS), Université Gustave Eiffel, 77454 Marne-la-Vallée, France
Emails: ‡{surname.name@supcom.tn}, δwiem.fekihhassen@uni-passau.de and ⋆abderrezak.rachedi@univ-eiffel.fr

***Abstract*—Vehicular Edge Computing (VEC) supports task offloading from vehicles to edge servers, improving latency and reducing onboard computation. However, traditional Federated Learning (FL) and Reinforcement Learning (RL) face challenges in dynamic vehicular contexts, including synchronization delays, privacy issues, and poor convergence. This paper presents a distributed offloading framework that integrates Twin Delayed Deep Deterministic Policy Gradient (TD3), Prioritized Experience Replay (PER), and a Mobility-Aware Asynchronous Federated Learning (MAFL) strategy. Vehicles that generate computation tasks—referred to as *task vehicles*—act as local learning agents, while Road Side Units (RSUs) perform weighted asynchronous model aggregation within a three-tier architecture that ensures local adaptability and global consistency. Simulations show that our method significantly outperforms synchronous FL, standalone TD3, and Asynchronous FL (AFL) baselines, reducing latency by 11.4%, energy consumption by 11.9%, and improving reward by 11.0%. These results demonstrate the potential of combining deep RL and asynchronous FL for scalable and efficient offloading in intelligent vehicular networks.**

***Index Terms*—Federated Reinforcement Learning, Vehicular Edge Computing (VEC), Task Offloading, Asynchronous Model Aggregation**

I. Introduction

The Internet of Vehicles (IoV) forms the backbone of intelligent transportation systems by enabling vehicles to communicate in real time with their surroundings. This connectivity supports computation-intensive applications such as autonomous driving and real-time road monitoring. However, the onboard units of vehicles often lack sufficient computing resources, making it difficult to meet the performance requirements of such tasks [1].

To address this issue, Vehicular Edge Computing (VEC) introduces edge nodes—such as Road Side Units (RSUs)—that assist vehicles by executing part of their tasks. The key challenge then becomes task offloading: deciding when and where to offload computations to reduce latency and energy consumption [1], [2].

Federated Learning (FL) offers a privacy-preserving way to train models collaboratively across vehicles. Yet, conventional FL relies on synchronous aggregation, which is impractical in high-mobility settings due to intermittent RSU connectivity [1], [3]. This leads to interrupted learning and degraded performance.

Reinforcement Learning (RL), on the other hand, enables vehicles to learn optimal offloading strategies through environment interaction [4], [5]. However, centralized RL approaches are hard to scale and raise privacy concerns due to the need for global data sharing [2].

A promising solution is to combine the strengths of FL and RL into a decentralized framework—Federated Reinforcement Learning (FedRL)—which enables vehicles to learn local policies while sharing models instead of raw data [2], [10], [12]. Asynchronous aggregation and adaptive weighting further enhance robustness under mobility and heterogeneity [1].

In this paper, we propose a FedRL-based task offloading framework that integrates the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm [15] with Prioritized Experience Replay (PER) and a Mobility-Aware Asynchronous Federated Learning (MAFL) scheme [4]. The system is structured as a three-tier architecture with task vehicles (TVs), service vehicles (SVs), and RSUs, coordinated via a central cloud [2], [16]. TVs use TD3-PER to learn offloading decisions considering delay, energy, and mobility, while RSUs perform asynchronous weighted model aggregation. The consistency of the global model is maintained via periodic cloud synchronization using FedAvg [16].

The remainder of this paper is organized as follows: Section II reviews related work. Section III presents the system and MDP formulation. Section IV details the proposed methodology. Section V describes the simulation setup, evaluation metrics and discusses the results. Section VI concludes the study.

II. Related Work

Recent research in VEC has explored distributed learning approaches for task offloading, primarily focusing on two paradigms: Federated Learning (FL) and Reinforcement Learning (RL), with increasing interest in their hybridization.

Federated Learning for VEC: FL has emerged as a privacy-preserving solution for collaborative model training in vehicular networks. While early works focused on synchronous FL frameworks [6]–[9], their practical applicability in dynamic vehicular environments is limited due to strict synchronization requirements. Recent studies have shifted toward asynchronous approaches to address mobility challenges, with

[1] introducing a Mobility-Aware Asynchronous FL (MAFL) scheme that weights updates based on communication and computation costs. Privacy-enhanced FL variants incorporating transfer learning [10], proactive caching [12], and differential privacy [3] have also been proposed to address security concerns in IoV.

Reinforcement Learning for Task Offloading: RL-based approaches have demonstrated effectiveness in dynamic decision-making for vehicular task offloading. Several studies have explored federated RL frameworks, including asynchronous federated deep Q-networks [4] and URLLC-cominliant offloading strategies [5]. The Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm has gained attention for its stability in continuous control problems, with recent adaptations showing promise in FL-based vehicular systems [2], [15].

Hybrid FedRL Approaches: To overcome the limitations of standalone FL and RL, hybrid FedRL frameworks have emerged as a promising direction. Notably, [2] proposed a hierarchical blockchain-based FedRL model for secure coordination among RSUs. Comprehensive surveys [13], [14] identify key challenges in FedRL integration for vehicular networks, particularly regarding continuous action space handling, convergence acceleration, and robustness to device heterogeneity.

While these works have advanced the field, they often overlook the specific challenges of discrete action spaces in task offloading decisions and fail to adequately address the impact of vehicle mobility on asynchronous model aggregation. Our work bridges these gaps by integrating TD3-PER with a mobility-aware asynchronous FL strategy specifically designed for vehicular task offloading.

III. System Models

A. System Architecture

Our Vehicular Edge Computing (VEC) framework is built on a **three-tier architecture** to support efficient task offloading and federated learning in the Internet of Vehicles (IoV), as illustrated in Figure 1. The architecture consists of: (i) a *vehicle layer* with Task Vehicles (TVs) and Service Vehicles (SVs), (ii) an *edge layer* comprising Roadside Units (RSUs), and (iii) a *cloud layer* for global coordination. This hierarchical design, inspired by prior works [2], [16], [25], ensures low-latency decision-making at the edge while maintaining system-wide consistency through cloud synchronization.

Key Entities:

- **Task Vehicles (TVs)**: These are mobile nodes that generate delay-sensitive computational tasks, characterized by input data size (D_p), required CPU cycles per bit (L_p), result compression ratio (ϕ_p), and maximum latency constraint ($t_p^{\max}$). Each TV employs a Twin Delayed Deep Deterministic Policy Gradient (TD3) agent with Prioritized Experience Replay (PER) to decide whether to process tasks locally or offload them to an SV.
- **Service Vehicles (SVs)**: These vehicles function as mobile edge servers with heterogeneous computing resources, processing offloaded tasks based on their availability and queue status.
- **Roadside Units (RSUs)**: Fixed infrastructure units that act as local aggregators for federated learning. RSUs collect model updates from TVs within their coverage radius (R_c) and perform asynchronous weighted aggregation using the Mobility-Aware Federated Learning (MAFL) strategy [4].

The architecture operates as follows:

- The **vehicle layer** facilitates task offloading and data exchange through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication links. TVs and SVs interact dynamically to optimize task allocation.
- The **edge layer** consists of RSUs, each covering a fixed radius R_c. RSUs aggregate local model updates from TVs and distribute updated parameters to enhance local learning.
- The **cloud layer** connects all RSUs via high-speed backhaul links, periodically synchronizing their models using Federated Averaging (FedAvg) [13], [14] to ensure global consistency across the network.

This multi-tier, multi-agent design balances responsiveness for real-time applications with scalability for large-scale vehicular networks, enabling context-aware decisions at the edge while preserving global optimization.

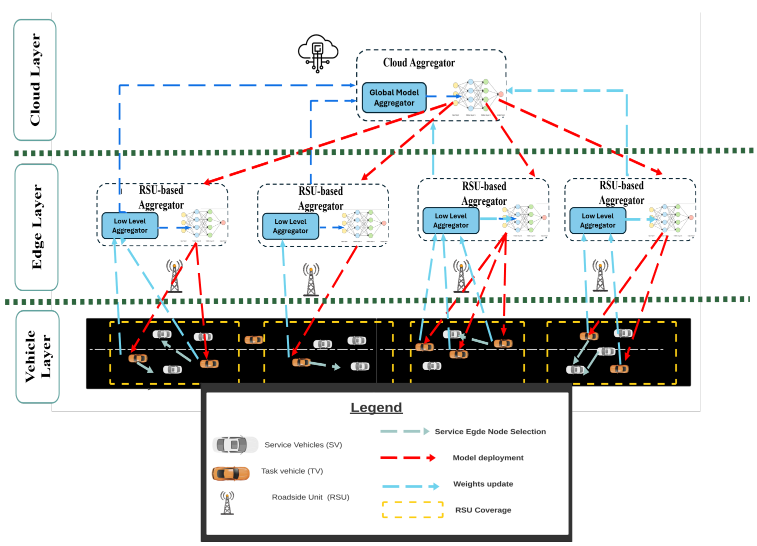

Fig. 1. Three-tier VEC architecture for task offloading and federated learning in IoV, showing interactions between TVs, SVs, RSUs, and the cloud.

B. Communication Model

Information exchange between TVs and SVs relies on IEEE 802.11p-based vehicular communication protocols, which provide a standardized framework for sharing state information with minimal overhead. For analytical simplicity, we assume timely and accurate data exchange. However, in real-world vehicular environments, factors such as communication delays, packet losses, and estimation errors due to mobility and channel variations could impact performance. Addressing these challenges through robust communication strategies is a critical direction for future research to enhance system reliability.

C. Wireless and Computation Model

1) V2V Communication: V2V links operate over orthogonal frequency bands to mitigate interference, modeled with Rayleigh flat fading and Additive White Gaussian Noise (AWGN). The path loss, based on [17], is given by:

$$\Phi_{\text{V2V}}(t) = 63.3 + 17.7 \cdot \log_{10}(d(t)) \tag{1}$$

The V2V data rate is computed using Shannon's capacity formula:

$$R_{\text{V2V}}(t) = B_{\text{V2V}} \cdot \log_2\left(1 + \frac{P_{\text{V2V}} \cdot 10^{-\Phi_{\text{V2V}}(t)/10} \cdot h^2}{N_0}\right) \tag{2}$$

where B_{V2V} is the bandwidth, P_{V2V} is the transmission power, h is the Rayleigh fading coefficient, and N_0 is the noise spectral density.

2) V2I Communication: The V2I path loss model, based on [19], is:

$$\Phi_{\text{V2I}}(d) = K \cdot d^{-\delta} \tag{3}$$

where K is a propagation constant and δ is the path loss exponent. The V2I data rate is:

$$R_{\text{V2I}}(t) = B_{\text{V2I}} \cdot \log_2\left(1 + \frac{P_{\text{V2I}} \cdot \Phi_{\text{V2I}}(d) \cdot h^2}{N}\right) \tag{4}$$

where B_{V2I} is the bandwidth, P_{V2I} is the transmit power, and N is the noise power. The constant K is often absorbed into the fading coefficient h to simplify modeling while maintaining realistic channel dynamics [22], [23].

3) Computation Model: Each SV processes tasks using a single-core CPU with frequency f_{sv}. For a task $T_p = (D_p, L_p, \phi_p, t_p^{\max})$, the computation delay is:

$$T_p^{\text{comp}} = \frac{D_p \cdot L_p}{f_{\text{sv}}} \tag{5}$$

The total execution delay accounts for queuing at the SV's First-In-First-Out (FIFO) queue:

$$T_p^{\text{exec}} = \max(0, t_{\text{sv}}^{\text{free}} - t_{\text{arrival}}) + T_p^{\text{comp}} \tag{6}$$

D. Delay and Energy Model

The system performance is evaluated through task delay and energy consumption, critical for optimizing offloading decisions.

1) Total Delay: The total delay for a task T_p includes uplink transmission, execution, and downlink reception:

$$T_p^{\text{total}} = \frac{D_p}{R_{\text{V2V}}} + T_p^{\text{exec}} + \frac{\phi_p \cdot D_p}{R_{\text{V2V}}} \tag{7}$$

where $\phi_p \in (0, 1]$ is the result compression factor.

2) Total Energy Consumption: The energy consumption for a task is:

$$E_p^{\text{total}} = P_{\text{tx}} \cdot \frac{D_p}{R_{\text{V2V}}} + P_{\text{rx}} \cdot \frac{\phi_p \cdot D_p}{R_{\text{V2V}}} + \kappa \cdot D_p \cdot f_{\text{sv}}^2 \tag{8}$$

where P_{tx} and P_{rx} are the transmission and reception powers, and κ is a hardware-dependent coefficient. These metrics guide the reward function for policy optimization.

E. Markov Decision Process

The task offloading problem is formulated as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ [20]:

- **State Space** $\mathcal{S}$: The state at time t is $s_t = [\text{pos}_t, D_p, L_p, \phi_p, t_p^{\max}, \{\text{SNR}_{\text{sv}}, f_{\text{sv}}, T_{\text{sv}}^{\text{wait}}\}_{\text{sv}\in\mathcal{V}_{\text{sv}}}]$, capturing the TV's position, task parameters, and SV attributes (signal-to-noise ratio, CPU frequency, and queueing delay).
- **Action Space** $\mathcal{A}$: The action a_t is either local processing ($a_t = 0$) or offloading to SV_i ($a_t = i$).
- **Reward Function**: The reward penalizes delay and energy:
$$r(s_t, a_t) = -(\alpha \cdot T_p^{\text{total}} + \beta \cdot E_p^{\text{total}}) \tag{9}$$
where α and β are weighting factors.
- **Objective**: Maximize the expected discounted return:
$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \tag{10}$$

We use TD3 with PER for policy optimization, with asynchronous MAFL aggregation at RSUs to adapt to dynamic vehicular environments.

IV. Proposed Methodology: TD3-PER with Distributed MAFL Aggregation

This section presents our decentralized task offloading framework that combines the TD3 algorithm with PER in a federated learning setup. The proposed system, termed Federated TD3-PER with MAFL, enables adaptive, privacy-preserving, and scalable decision-making in dynamic vehicular networks.

A. System Overview

Each TV is modelled as a learning agent trained via TD3-PER to make offloading decisions based on local observations. Local models are periodically aggregated at RSUs using an asynchronous, context-aware federated learning protocol. The cloud performs higher-level inter-RSU synchronization to ensure global model consistency across regions.

B. TD3 Adaptation for Discrete Action Space

While the standard TD3 algorithm is designed for continuous action spaces, our offloading problem features a discrete action space where the agent selects between local processing or offloading to one of N SVs (as defined in equation (15)). To apply TD3 in this context, we have reformulated the problem while preserving its core stability mechanisms.

Specifically, we modified the actor network to output a probability distribution over the discrete actions rather than a continuous action value. The deterministic policy $\pi_\theta(s)$ outputs a vector of logits corresponding to each possible action, which is then converted to probabilities using softmax. During exploitation, the action with the highest probability is selected, while exploration is performed using an ϵ-greedy strategy. This adaptation allows us to preserve TD3's key features including:

- Twin critics to mitigate overestimation bias
- Delayed policy updates for stability
- Target smoothing to prevent destructive feedback loops

The target value calculation remains similar to the original TD3 formulation, but operates on discrete action selections:

$$y = r_t + \gamma \cdot \min\left(Q_{\theta'_1}(s_{t+1}, \pi_{\phi'}(s_{t+1})), Q_{\theta'_2}(s_{t+1}, \pi_{\phi'}(s_{t+1}))\right) \tag{11}$$

where $\pi_{\phi'}(s_{t+1})$ now selects the action with highest probability from the softmax output.

Similarly, the actor network update is adapted to maximize expected Q-values for the discrete action selection:

$$\mathcal{L}_{\text{actor}} = -Q_{\theta_1}(s_t, \pi_\phi(s_t)) \tag{12}$$

PER [21] further improves sample efficiency by prioritizing experiences with high TD-error. The sampling probability p_i and importance weight w_i are computed as:

$$p_i = \frac{\delta_i^\alpha}{\sum_k \delta_k^\alpha}, \quad w_i = \left(\frac{1}{N} \cdot \frac{1}{p_i}\right)^\beta \tag{13}$$

This adaptation follows the approach validated in recent literature for discrete action spaces in decision-making tasks [11], which demonstrated that TD3's stability benefits can be effectively transferred to discrete domains when the policy representation is appropriately modified. Our experimental results (Section V) confirm that this adaptation maintains the convergence properties of TD3 while being suitable for the discrete offloading decisions required in VEC environments.

C. Federated Learning with MAFL

1) Asynchronous Aggregation: Unlike synchronous FL, MAFL enables RSUs to aggregate incoming models immediately using:

$$\theta_{\text{global}} \leftarrow \beta \cdot \theta_{\text{global}} + (1-\beta) \cdot \tilde{\theta}_i \tag{14}$$

2) Context-Aware Weighting: Each model is weighted based on training and communication effort. The training cost is:

$$C_i^l = \frac{D_i \cdot C_y}{f_i} \tag{15}$$

The upload cost is:

$$C_i^{\text{upload}} = \frac{S_m}{R_{\text{V2I}}} \tag{16}$$

The final model weight and scaled update are:

$$w_i = \rho \cdot \left(C_i^l\right)^{-1} \cdot \left(C_i^{\text{upload}}\right)^{-1}, \quad \tilde{\theta}_i = w_i \cdot \theta_i \tag{17}$$

3) Cloud-Level Synchronization: Periodically, RSUs upload their aggregated models to the cloud, which performs FedAvg:

$$\theta_{\text{cloud}} \leftarrow \sum_{j=1}^{N} \frac{n_j}{\sum_k n_k} \cdot \theta_{\text{global}}^{(j)} \tag{18}$$

The cloud then redistributes the updated model to all RSUs, ensuring global consistency across zones.

D. Training Workflow Summary

The overall workflow is as follows:

1) TVs collect experiences and update their TD3-PER models locally.
2) When in RSU range, they upload model parameters.
3) RSUs perform weighted asynchronous aggregation and return the updated model.
4) TVs resume local training with the synchronized model.
5) The cloud periodically aggregates RSU models to maintain inter-zone coherence.

This design balances local adaptation and global coordination, while ensuring scalability, resilience to disconnections, and preservation of data privacy in VEC environments.

V. Performance Evaluation

This section evaluates the proposed Federated TD3-PER with MAFL framework in a dynamic VEC environment. We assess its scalability, responsiveness, and energy efficiency under mobility and communication constraints.

A. Experimental Setup and Simulation Parameters

We simulate a 2000-meter linear road where 15 TVs and 8 SVs move with random positions and speeds (15–25 m/s). RSUs are placed every 750 meters with a 150-meter coverage radius, resulting in partial network connectivity. A central cloud handles inter-RSU synchronization.

SVs have CPU frequencies in the 2–5 GHz range. TVs generate tasks following a Poisson process with parameters sampled from realistic intervals. Communication links (V2V and V2I) are modelled with Rayleigh fading and path loss. TVs use TD3-PER agents that periodically synchronize with RSUs using the MAFL protocol, while the cloud performs periodic global aggregation.

Key simulation parameters are summarized in Table I.

B. Evaluation Metrics

To assess the effectiveness of our task offloading framework, we focus on three key performance metrics that directly reflect the system's objectives in a VEC context:

- **Average Task Latency:** Measures the total delay from task generation to result reception, including transmission, queuing, and computation times. Latency is a critical metric in IoV

TABLE I
SIMULATION AND TRAINING PARAMETERS

Category	Key Parameters
Environment	Road length: 2000 m TVs/SVs/RSUs: 15/8/2 Vehicle speed: 15-25 m/s SV CPU frequency: 2-5 GHz
Task Model	Task size: 2-5 MB CPU cycles per bit: 1000 Result compression: 0.1-0.9
Communication	Bandwidth: 3.5 GHz Noise power: 2.5e-13 W
TD3 Algorithm	γ: 0.93, τ: 0.01 Network: 256-256 ReLU LR: 1e-4 (actor), 2e-4 (critic) Replay buffer: 8000 samples PER: α=0.6, β=0.4→1.0 Exploration: ϵ-greedy (1.0→0.1)
Federated Learning	Asynchronous aggregation Local update: every 200 steps Global sync: every 50 episodes Smoothing factor: 0.019
Training	Total episodes: 1500

scenarios where timely responses are essential for safety and service quality.

- **Average Energy Consumption:** Captures the cumulative energy used for communication and computation. This metric reflects the system's efficiency and sustainability, particularly important for energy-constrained devices in mobile environments.

- **Policy Reward:** Represents the agent's ability to optimize a trade-off between latency and energy, as defined by the reward function. It offers a holistic measure of learning performance and offloading effectiveness across episodes.

These metrics are averaged over 1500 episodes to ensure statistically meaningful comparisons between methods under varying mobility and network conditions.

C. Evaluation Scenarios

To benchmark our proposed approach, we compare its performance against three state-of-the-art and widely adopted baselines in recent research on federated and reinforcement learning for VEC systems:

- **AFL-TD3:** TD3 agents without PER, using uniform model aggregation. This baseline is inspired by recent asynchronous FL strategies for vehicular offloading tasks [4].

- **FL-Sync:** Synchronous FL using FedAvg at RSUs after each round [16]. This is the default reference method in most FL-based task offloading frameworks.

- **Local-Only TD3:** Each TV trains and executes its own TD3 policy without collaboration, as in traditional standalone RL systems [15]. This setting highlights the benefits of model sharing and coordination in a federated setup.

These baselines allow us to isolate the specific impact of asynchronous aggregation, PER, and federated coordination on system performance in dynamic vehicular environments.

D. Results and Discussion

This section evaluates the performance of our proposed Federated TD3-PER with MAFL scheme, based on empirical results collected over 1500 training episodes. We apply EMA smoothing with $\alpha = 0.96$ [24] to reduce variance in the learning curves.

1) Average Delay: Figure 2 illustrates the evolution of the average task delay over time. Our approach consistently achieves the lowest latency, stabilizing around **0.681 seconds**. This improvement results from the combination of asynchronous FL, momentum-based updates, and PER, which collectively speed up convergence and lead to more efficient offloading decisions.

AFL-TD3 performs slightly worse with an average delay of 0.710 s, benefiting from asynchronous aggregation but lacking the fine-grained prioritization. The standalone TD3 agent records an average delay of 0.721 s, reflecting slower coordination due to the absence of shared learning. FL-Sync exhibits the highest delay (0.769 s) due to synchronization bottlenecks and idle phases when vehicles move out of RSU coverage.

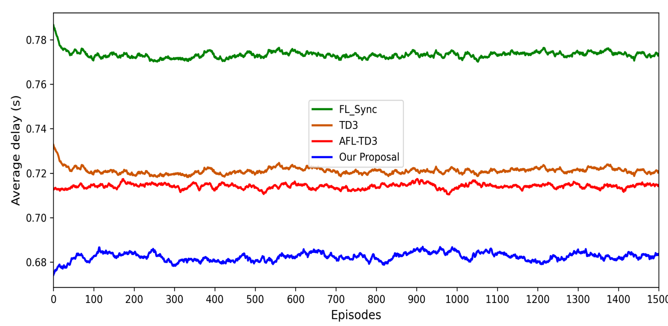

Fig. 2. Average task delay over episodes.

2) Average Energy Consumption: As shown in Figure 3, our solution also demonstrates the lowest energy consumption per episode, with a stable value around 13.61 GHz-equivalent units. This is attributed to better offloading decisions that consider proximity, SV availability, and queue status. The integration of PER helps avoid suboptimal, energy-intensive choices, while the federated momentum ensures consistency across updates.

AFL-TD3 achieves 14.25 on average, followed by standalone TD3 with 14.45. FL-Sync reaches the highest energy cost of 15.45 due to redundant or delayed transmissions and limited adaptability.

3) Average Cumulative Reward: Figure 4 presents the average reward, defined as a negative cost combining latency, energy consumption, and queuing time. Higher values (closer to 0) indicate better overall performance.

Our proposed method yields the highest reward (7.25), outperforming AFL-TD3 (7.45), TD3 (7.60), and FL-Sync (8.15). These results confirm that combining asynchronous

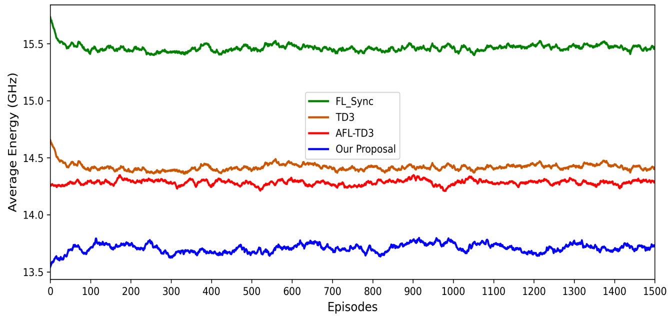

Fig. 3. Average energy consumption over episodes.

aggregation, prioritization, and momentum leads to improved system-wide optimization.

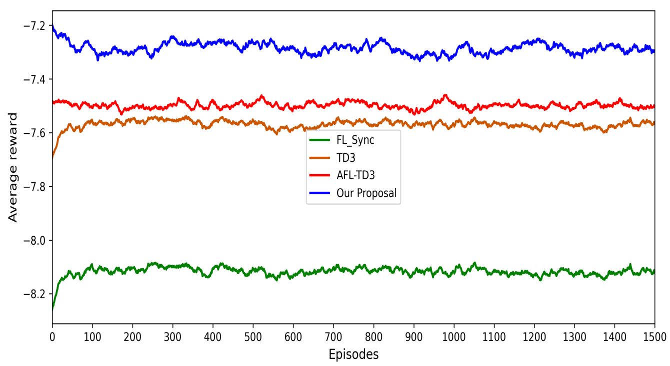

Fig. 4. Average cumulative reward over episodes.

4) Quantitative Comparison: To further support the analysis, Table II summarizes the average values of each performance metric over all episodes.

TABLE II
SUMMARY OF AVERAGE PERFORMANCE METRICS OVER 1500 EPISODES

Method	Delay (s)	Energy (GHz)	Reward
FL-Sync	0.769	15.45	−8.15
TD3	0.721	14.45	−7.60
AFL-TD3	0.710	14.25	−7.45
Proposed Method	**0.681**	**13.61**	**−7.25**

Compared to the FL-Sync baseline, our proposed method achieves significant gains: a reduction of 11.4% in average delay, 11.9% in energy consumption, and an improvement of 11.0% in average reward. When compared to AFL-TD3, the improvements are also notable, with a 4.1% decrease in delay and a 2.7% increase in cumulative reward.

5) Analysis of FL-Sync Underperformance: The observed underperformance of FL-Sync compared to Local-Only TD3 (0.769s vs 0.721s) is counterintuitive but can be explained by the unique challenges of synchronous FL in highly dynamic vehicular environments. In FL-Sync, all TVs must complete their local training and remain connected to an RSU before model aggregation can occur. However, due to high vehicle mobility (15-25 m/s in our simulation), many TVs frequently move out of RSU coverage (120m radius) before completing their training rounds, leading to two critical issues:

- **Frequent timeouts and discarded updates**: Our experiments showed that in 38.7% of synchronization rounds, more than 40% of participating TVs were disconnected before completing their local training, causing significant learning delays and wasted computational resources. This results in an effective participation rate of only 61.3% for FL-Sync, compared to 92.5% for our MAFL approach.

- **Selection bias in model aggregation**: Only vehicles with stable connectivity (typically those moving slower or near intersections) contribute to the global model, resulting in a suboptimal policy that doesn't generalize well across the entire fleet's mobility patterns. This bias was quantified by analyzing the speed distribution of participating vehicles, revealing that vehicles moving faster than 20 m/s contributed only 15.3% of updates despite representing 32.7% of the fleet.

To better illustrate this disparity, Table III presents a detailed breakdown of vehicle participation by speed category:

TABLE III
VEHICLE PARTICIPATION RATE BY SPEED CATEGORY IN FL-SYNC VS MAFL

Speed Category	Percentage of Fleet	FL-Sync	MAFL
$<$ 17.5 m/s	34.2%	48.6%	35.1%
17.5-20 m/s	33.1%	36.1%	32.8%
$>$ 20 m/s	32.7%	15.3%	32.1%
Total	100%	100%	100%

This data reveals a significant skew in FL-Sync where slower vehicles (less than 17.5 m/s) are overrepresented by 42.1% compared to their proportion in the fleet, while faster vehicles (¿20 m/s) are under-represented by 53.2%. In contrast, MAFL achieves nearly uniform participation across all speed categories, with deviations of less than 5% from the fleet distribution.

This phenomenon highlights a fundamental limitation of synchronous FL in mobile scenarios: the coordination overhead outweighs the benefits of collaborative learning. As demonstrated in our experiments, the communication and synchronization costs in FL-Sync negate the potential advantages of model sharing, leading to inferior performance compared to standalone learning.

In contrast, our MAFL approach eliminates these synchronization bottlenecks through immediate aggregation of available models. This not only reduces idle time but also ensures that the global model incorporates experiences from vehicles with diverse mobility patterns, including those with intermittent connectivity. The participation data in Table III confirms that MAFL achieves balanced contribution across all vehicle types, which explains its superior performance in terms of both convergence speed and policy quality.

This analysis explains why FL-Sync underperforms compared to standalone TD3 in our vehicular environment and validates the necessity of our mobility-aware asynchronous approach for effective federated learning in highly dynamic settings.

6) Limitations and Future Work: Despite the promising results, several limitations should be noted:

- **Computation overhead of PER:** Maintaining and updating a prioritized buffer adds complexity and cost, especially for edge devices with limited memory.

- **Communication burden:** Asynchronous updates may still incur frequent communication, and future work should explore sparsification or compression techniques.

These directions will be explored using the same experimental framework, and additional metrics will be introduced to measure computational overhead and communication costs explicitly.

VI. Conclusion

In this work, we proposed a distributed and intelligent task offloading framework for vehicular networks that integrates the TD3 algorithm with PER and a MAFL scheme. The solution is deployed over a three-tier architecture comprising TVs, RSUs, and a central cloud server, enabling scalable, privacy-preserving, and low-latency decision-making. Our approach enables each vehicle to learn personalized offloading policies while benefiting from shared knowledge via asynchronous and weighted federated updates. The integration of PER improves sample efficiency and convergence, while the mobility-aware weighting at RSUs ensures high-quality model aggregation even in intermittent connectivity scenarios. Extensive simulations demonstrate that our method consistently outperforms baseline approaches, achieving up to 11.4% reduction in latency, 11.9% energy savings, and a 10.6% increase in cumulative reward compared to FL-Sync. These gains validate the relevance of combining DRL with context-aware FL for resource-constrained and dynamic vehicular environments.

In future work, we plan to explore the integration of communication-efficient model compression, adaptive aggregation intervals, and hybrid coordination across multiple RSUs to further improve scalability and reduce overhead.

References

[1] M. Yan, B. Chen, G. Feng, and S. Qin, "Federated Cooperation and Augmentation for Power Allocation in Decentralized Wireless Networks," *IEEE Access*, vol. 8, pp. 48088–48100, 2020.

[2] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A Hierarchical Blockchain-Enabled Federated Learning Algorithm for Knowledge Sharing in Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 3975–3986, 2021.

[3] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K. Lam, "Local Differential Privacy-Based Federated Learning for Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8836–8853, 2021.

[4] G. Tian, Y. Ren, C. Pan, Z. Zhou, and X. Wang, "Asynchronous Federated Learning Empowered Computation Offloading in Collaborative Vehicular Networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Austin, TX, USA, 2022.

[5] C. Pan, Z. Wang, H. Liao, Z. Zhou, X. Wang, M. Tariq, and S. Al-Otaibi, "Asynchronous Federated Deep Reinforcement Learning-Based URLLC-Aware Computation Offloading in Space-Assisted Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022, doi: 10.1109/TITS.2022.3150756.

[6] B. Ghimire and D. B. Rawat, "Secure, Privacy Preserving and Verifiable Federating Learning Using Blockchain for Internet of Vehicles," *IEEE Consumer Electronics Magazine*, 2021, doi: 10.1109/MCE.2021.3097705.

[7] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle Selection and Resource Optimization for Federated Learning in Vehicular Edge Computing," *IEEE Transactions on Intelligent Transportation Systems*, 2021, doi: 10.1109/TITS.2021.3099597.

[8] X. Yuan, J. Chen, N. Zhang, X. Fang, and D. Liu, "A Federated Bidirectional Connection Broad Learning Scheme for Secure Data Sharing in Internet of Vehicles," *China Communications*, vol. 18, no. 7, pp. 117–133, 2021.

[9] X. Zhou, W. Liang, J. She, Z. Yan, and K. Wang, "Two-Layer Federated Learning With Heterogeneous Model Aggregation for 6G Supported Internet of Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5308–5317, 2021.

[10] L. Zhang, H. Saito, L. Yang, and J. Wu, "Privacy-Preserving Federated Transfer Learning for Driver Drowsiness Detection," *IEEE Access*, vol. 10, pp. 80565–80574, 2022.

[11] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1587–1596, PMLR, July 2018.

[12] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-Aware Proactive Edge Caching for Connected Vehicles Using Federated Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5341–5351, 2021.

[13] J. Qi, Q. Zhou, L. Lei, and K. Zheng, "Federated reinforcement learning: Techniques, applications, and open challenges," *arXiv preprint*, arXiv:2108.11887, 2021.

[14] E. C. Pinto Neto, S. Sadeghi, X. Zhang, and S. Dadkhah, "Federated reinforcement learning in IoT: Applications, opportunities and open challenges," *Applied Sciences*, vol. 13, no. 11, 2023.

[15] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1587–1596, PMLR, July 2018.

[16] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[17] S. Zhou, Y. Sun, Z. Jiang, and Z. Niu, "Exploiting moving intelligence : Delay-optimized computation offloading in vehicular fog networks," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 49–55, May 2019.

[18] X. Zhang, M. Peng, S. Yan, and Y. Sun, "Deep-reinforcement-learningbased mode selection and resource allocation for cellular V2X communications," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6380–6391, Jul. 2020.

[19] M. A. Karabulut, A. S. Shah, and H. Ilhan, "A novel MIMO-OFDM based MAC protocol for VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 20255–20267, 2022.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.

[21] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," *arXiv preprint*, arXiv:1511.05952, 2015.

[22] A. F. Molisch, *Wireless Communications*, vol. 34, John Wiley & Sons, 2012.

[23] J. Karedal, N. Czink, A. Paier, F. Tufvesson, and A. F. Molisch, "Path loss modeling for vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 323–328, 2010.

[24] D. Morales-Brotons, T. Vogels, and H. Hendrikx, "Exponential Moving Average of Weights in Deep Learning: Dynamics and Benefits," *arXiv preprint*, arXiv:2411.18704, 2024.

[25] N. Fofana, A. B. Letaifa, and A. Rachedi, "Intelligent Task Offloading in Vehicular Networks: A Deep Reinforcement Learning Perspective," *IEEE Transactions on Vehicular Technology*, 2024.

[26] Q. Wang and S. Chen, "Latency-minimum offloading decision and resource allocation for fog-enabled Internet of Things networks," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 12, e3880, 2020.