# Poster: Simulator for Reinforcement Learning-Based Resource Management in Vehicular Edge Computing

Agon Memedi*, Chunghan Lee†, Seyhan Ucar†, Onur Altintas†, and Falko Dressler*
*School of Electrical Engineering and Computer Science, TU Berlin, Germany
†InfoTech Labs, Toyota Motor North America R&D, CA, U.S.A.
{memedi,dressler}@ccs-labs.org,{chunghan.lee1,seyhan.ucar,onur.altintas}@toyota.com

***Abstract*—We aim to investigate Reinforcement Learning (RL) methods for efficient resource management in Vehicular Edge Computing (VEC). To this end, we present an open-source, modular, lightweight, discrete-event simulation framework which integrates state-of-the-art tools for improved performance evaluation. By integrating realistic mobility traces, our approach presents an opportunity to evaluate the performance and scalability of different RL-based task scheduling and resource allocation policies in diverse scenarios. This offers flexibility and insights into the generalizability of RL-based scheduling policies. We make the framework available as open-source to foster broader accessibility, support research in the field. We present early results to demonstrate the potential of this simulator.**

## I. Introduction

The proliferation of computationally intensive applications, such as Generative AI, Virtual Reality, and Augmented Reality, has increased the demand for more computing and networking resources [1]. While traditional distributed computing infrastructures (e.g., cloud computing and edge computing) are readily available, their scalability is hindered by high latency and bandwidth limitations (in cloud computing); and server maintenance and deployment costs (in edge computing). These challenges have sparked interest toward novel resource infrastructures that extend beyond traditional server deployments.

One emerging paradigm is Virtual Edge Computing (V-Edge) [2], which proposes the use of computational power of everyday devices to form a unified resource pool, independent of the fixed infrastructure. In the vehicular domain, this paradigm is conceptualized in the form of Vehicular Micro Clouds (VMCs) [3]. VMCs represent a cluster of stationary or mobile vehicles with processing, memory, storage, and communication resources that are available on-demand, therefore enabling Vehicular Edge Computing (VEC).

Due to the inherent mobility of vehicles, however, the VEC exhibits dynamic behavior with fluctuating availability of resources. This variability imposes challenges with respect to resource management and utilization. As such, the prediction of resource availability and demand, and effective allocation of resources becomes essential for maintaining system stability and performance. Machine Learning (ML) approaches, particularly Reinforcement Learning (RL), emerge as a feasible solution for addressing these challenges due to their ability to learn and adapt in complex, dynamic environments. By leveraging historical data and real-time information, these approaches offer a proactive means to manage resources, ultimately leading to more resilient and efficient VEC infrastructure.

RL-based approaches, in particular Deep Reinforcement Learning (DRL), have already been used in the literature [4]–[7]. For instance, Guo et al. [4] and Binh et al. [5] use DRL techniques to optimize task offloading decisions in VEC. Similarly, Liu et al. [6] combine DRL and Directed Acyclic Graphs (DAGs) to allocate interdependent subtasks to vehicles, optimizing the overall task completion time. Wu et al. [7] introduce *VEC-Sim*, which uses DRL to model the mobility trajectories and driving behaviors of vehicles. Nevertheless, many studies rely on simplified models, overlooking the impact of real-world scenarios on RL systems' behavior. While some works offer a more comprehensive approach, their proprietary nature restricts opportunities for further research.

To address these challenges, adequate tooling support is needed. Schettler et al. [8] studied how to integrate RL into the popular Veins simulation framework. Despite the advantages, simulation performance and flexibility were limited, particularly when considering distributed and edge computing applications.

In this paper, we propose a novel simulation framework designed specifically for RL-based task scheduling and resource allocation in VEC. Key features include: (1) Realistic Modeling: The framework integrates authentic vehicle mobility traces, ensuring an accurate representation of road traffic behavior; (2) Flexibility for ML Research: It enables rigorous investigation of RL policies across diverse VMCs scenarios; (3) Open Source Access: We release the simulation framework as open-source, fostering further research and collaboration in the field.[1]

## II. Simulation Environment and Components

Our simulation framework integrates SimPy[2], a process-based discrete-event simulation library, with SUMO[3], a microscopic traffic simulator, to simulate road traffic flow and vehicle mobility. Furthermore, the PyTorch machine learning framework is incorporated to develop deep learning models for RL algorithms. Together, these components facilitate the investigation of complex and diverse scenarios that closely mimic real-world conditions offering insights into the performance and scalability of RL-based solutions.

[1] https://github.com/agnmmd/ve-sim
[2] https://simpy.readthedocs.io/en/latest/
[3] https://eclipse.dev/sumo/

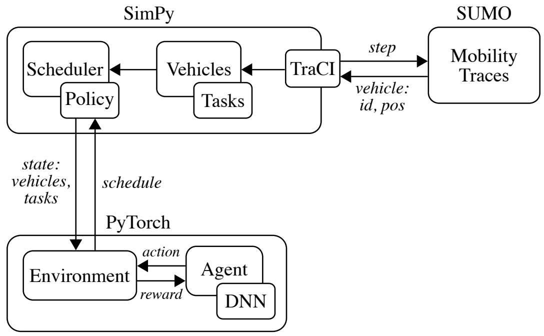


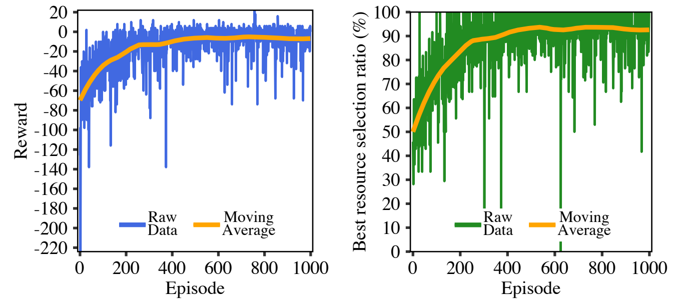

Figure 1. Main components of the simulation framework.

Figure 1 shows the relevant system-model components of the simulation framework. The main system-model components in the simulator include *Vehicles*, *Tasks* and *Scheduler*. Vehicles are modeled as mobile compute resources with certain processing capacity that can both generate and process tasks. The mobility of the vehicles is determined by the simulated SUMO scenario, whose traces are fed to our simulation environment through the TraCI interface. Tasks, on the other hand, present a computational workload that require processing resources. They are defined by attributes such as complexity, deadline, and priority, which can be used as scheduling criteria. Tasks can either be processed locally, by the generating vehicle, or migrated to another vehicle. The scheduler plays the central role for determining task assignment and resource allocation. It has an overview of the scenario state (available tasks and resources) and matches tasks and resources following a certain *Policy*. The policy essentially specifies the criteria for selecting the next task and resource to be matched and scheduled. These criteria can be based on heuristic methods (e.g., prioritizing the task with the earliest deadline or lowest complexity first, or in the case of vehicles, the vehicle with the highest processing capacity or dwell time), and RL-based algorithms. For RL-based policies, the current system state, comprising of information about vehicles and tasks, is fed to the *Environment* component. The actual *observation space* is a vector that combines task information and resource capabilities (i.e., count of available resources, complexity and deadline of the selected task, processing power of each available resource.) The *action space* consist of the currently available resources.

An *Agent*, implementing a specific RL algorithm, then interacts with the Environment to learn optimal decision-making strategies for matching tasks with appropriate resources, guided by a predefined *reward function*. The agent receives a positive reward when it selects the fastest available resource that can complete the task before its deadline; neutral feedback when it correctly selects the fastest resource, regardless of the deadline; and it is strongly penalized when it doesn't select the fastest available resource.

## III. Preliminary Simulation Results

To demonstrate the feasibility of our approach, we implemented a Deep Q-Learning (DQN) algorithm for the RL agent. The agent's task is to select the best resource for processing a given task, and eventually learn the optimal task-resource matching. The agent is rewarded when the resource selection leads to a successful task completion before the deadline, and punished otherwise. We evaluated the system in a scenario where tasks and resources are dynamically generated, modeling real-world variability. Furthermore, we set the parameters in our simulation (i.e., task's complexity, deadlines and arrival rate and resource's processing capacity) to model a VEC with limited resources and time-sensitive tasks.




(a) Collected rewards per episode (blue) and moving average (orange).

(b) Ratio of selecting the best resource in an episode's action space (green).

Figure 2. Learning performance of the RL agent.

Figure 2 shows the performance of the implemented RL-based policy, indicating agent's ability to learn to select the best available resource (Figure 2b) guided by the reward function (Figure 2a). Figure 2a shows the convergence of the reward at about 250 episodes. This leads to excellent resource selection (cf. Figure 2b). The results demonstrate the framework's potential to support RL-based task scheduling and resource allocation in dynamic VEC environments.

## References

[1] C. E. Casetti, C. F. Chiasserini, F. Dressler, A. Memedi, D. Gasco, and E. M. Schiller, "AI/ML-based Services and Applications for 6G-Connected and Autonomous Vehicles," *Elsevier Computer Networks*, vol. 255, p. 110 854, Dec. 2024.

[2] F. Dressler, C. F. Chiasserini, F. H. P. Fitzek, H. Karl, R. Lo Cigno, A. Capone, C. E. Casetti, F. Malandrino, V. Mancuso, F. Klingler, and G. A. Rizzo, "V-Edge: Virtual Edge Computing as an Enabler for Novel Microservices and Cooperative Computing," *IEEE Network*, vol. 36, no. 3, pp. 24–31, May 2022.

[3] T. Higuchi, J. Joy, F. Dressler, M. Gerla, and O. Altintas, "On the Feasibility of Vehicular Micro Clouds," in *IEEE VNC 2017*, Turin, Italy: IEEE, Nov. 2017, pp. 179–182.

[4] H. Guo, J. Liu, J. Ren, and Y. Zhang, "Intelligent Task Offloading in Vehicular Edge Computing Networks," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 126–132, Aug. 2020.

[5] T. H. Binh, D. B. Son, H. Vo, B. M. Nguyen, and H. T. T. Binh, "Reinforcement Learning for Optimizing Delay-Sensitive Task Offloading in Vehicular Edge–Cloud Computing," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 2058–2069, Nov. 2020.

[6] Z. Liu, L. Huang, Z. Gao, M. Luo, S. Hosseinalipour, and H. Dai, "GA-DRL: Graph Neural Network-Augmented Deep Reinforcement Learning for DAG Task Scheduling Over Dynamic Vehicular Clouds," *IEEE Transactions on Network and Service Management*, vol. 21, no. 4, pp. 4226–4242, Aug. 2024.

[7] F. Wu, X. Xu, M. Bilal, X. Wang, H. Cheng, and S. Wu, "VEC-Sim: A simulation platform for evaluating service caching and computation offloading policies in Vehicular Edge Networks," *Elsevier Computer Networks*, vol. 257, 2025.

[8] M. Schettler, D. S. Buse, A. Zubow, and F. Dressler, "How to Train your ITS? Integrating Machine Learning with Vehicular Network Simulation," in *IEEE VNC 2020*, Virtual Conference: IEEE, Dec. 2020.