Developing an Eco-Driving Strategy in a Hybrid Traffic Network Using Reinforcement Learning

Umar Jamil[1], Mostafa Malmir[1], Alan Chen[2], Monika Filipovska[3], Mimi Xie[4], Caiwen Ding[5], Yu-Fang Jin[1§]

[1]Department of Electrical and Computer Engineering, the University of Texas at San Antonio, San Antonio, TX 78249, USA

[2]Westlake High school, Austin TX 78746, USA

[3]Department of Civil Engineering, University of Connecticut, Storrs, CT 06269, USA

[4]Department of Computer Science, the University of Texas at San Antonio, San Antonio, TX 78249, USA

[5]Department of Computer Science & Engineering, University of Connecticut, Storrs, CT 06269, USA

[§]**Correspondence** should be addressed to Yu-Fang Jin (yufang.jin@utsa.edu)

Abstract

Eco-driving has garnered considerable research attention owing to its potential socio-economic impact, including enhanced public health and mitigated climate change effects through the reduction of greenhouse gas emissions. With an expectation of more autonomous vehicles (AV) on the road, an eco-driving strategy in hybrid traffic networks encompassing AV and human-driven vehicles (HDV) with the coordination of traffic lights is a challenging task. The challenge is partially due to the insufficient infrastructure for collecting, transmitting, and sharing real-time traffic data among vehicles, facilities, and traffic control centers, and the following decision-making of agents involved in traffic control. Additionally, the intricate nature of the existing traffic network, with its diverse array of vehicles and facilities, contributes to the challenge by hindering the development of a mathematical model for accurately characterizing the traffic network. In this study, we utilized the Simulation of Urban Mobility (SUMO) simulator to tackle the first challenge through computational analysis. To address the second challenge, we employed a model-free reinforcement learning (RL) algorithm, Proximal policy optimization (PPO), to decide the actions of AV and traffic light signals in a traffic network. A novel eco-driving strategy was proposed by introducing different percentages of AV into the traffic flow and collaborating with traffic light signals using RL to control the overall speed of the vehicles, resulting in improved fuel consumption efficiency. Average

rewards with different penetration rates of AV (5%, 10%, and 20% of total vehicles) were compared to the situation without any AV in the traffic flow (0% penetration rate). The 10% penetration rate of AV showed a minimum time of convergence to achieve average reward, leading to a significant reduction in fuel consumption and total delay of all vehicles.

Keywords: Eco-driving; Hybrid Traffic Network; Reinforcement Learning; Traffic Flow Control; Fuel Consumption; Microscopic Traffic Simulator

1. Introduction

Findings from a 2022 study indicate that the transportation sector accounted for 27% of the energy consumption in the United States.[1] Specifically, petroleum (gasoline) consumption comprised about 52% of the total energy consumption, resulting in significant air pollutant emissions. This underscores the necessity for a well-designed traffic control system to mitigate fuel energy consumption (FEC) and air pollution for sustainability.[2-4] The concept of sustainability has driven research into eco-driving strategies designed to reduce FEC rates (FEC within time). FEC rates can be calculated based on factors such as acceleration, mass, drag coefficient, rolling coefficient, driveline efficiency, idling speed, and idling fuel mean pressure.[5,6] Reducing FEC involves two interconnected goals: shorter travel time and lower FEC rates. Vehicles incur the highest

FEC rates during idling and frequent stops and starts, especially at traffic lights or in congestion. Therefore, prioritizing the establishment of a continuous traffic flow, characterized by minimal fluctuations in vehicle speeds, is essential for achieving lower FEC rates and shorter traffic delays. This approach is instrumental in promoting effective eco-driving strategies.[7]

Traditional traffic control relies on fixed modes for traffic light changes and manual rerouting, resulting in limited efficiency and a lack of feedback mechanisms. The current setup of traffic control systems poses challenges in developing eco-driving strategies for hybrid traffic networks encompassing AV and HDV. These challenges stem partially from the insufficient infrastructure for collecting, transmitting, and sharing real-time traffic data among vehicles, facilities, and traffic control centers, as well as the subsequent decision-making by involved agents. Furthermore, the intricate nature of the existing traffic networks, with their diverse array of vehicles and facilities, complicates the development of a mathematical model for accurately characterizing the traffic networks.

Current eco-driving strategies have addressed the challenges from various perspectives, including real-time artificial intelligence for traffic monitoring, and 5th generation (5G) communication networks to facilitate rapid information sharing.[8-10] Due to the multifaceted nature of the eco-driving problem, a model-based deterministic strategy is challenging to approach. Meanwhile, data-driven approaches show promise, given the large amount of data accumulated during the past decades.

Related Work on Reinforcement Learning in Traffic Control

Model-free RL has demonstrated its advantage in decision-making for traffic flow control by examining interactions among multiple agents and the environment.[10-12] RL has been applied to optimize vehicle routes for reduced delay and vehicle accelerations for less FEC.[13,14] RL algorithms have also been developed to reduce air pollutant emissions by reducing vehicles' waiting time at road intersections.[15,16] In a study on infrastructure-to-vehicle communications networks,[17] a single vehicle was considered as an agent, and the Q-learning (QL) algorithm was developed to minimize carbon dioxide emissions. Additionally, a recent eco-driving framework based on the deep Q-network (DQN) approach was presented to enhance the fuel efficiency of multiple vehicles in a traffic network with one horizontal road and one vertical road.[18]

In addition to applications of RL in controlling vehicle routes or acceleration, traffic lights are also considered as agents to control traffic flow with RL algorithms. An RL-based control has been developed for smart traffic signals, to reduce traffic jams and improve traffic smoothness in a traffic grid consisting of 3 horizontal and 3 vertical roads.[19]

With more AV running on the road, they are also considered agents in RL algorithms for traffic control. In a recent study, a circular network with fixed traffic signal patterns at one spot was deployed to develop a deep deterministic policy gradient (DDPG) algorithm. The study aims to minimize the FEC of Connected AV by controlling their acceleration.[20] Additionally, RL algorithms with a hybrid deep Q-learning and policy gradient (HDQPG) were developed to minimize the FEC of Connected AV by controlling their acceleration in a traffic grid with one horizontal and five vertical roads.[21] Previous studies also explored a traffic flow containing both HDV and Connected AV using a trust region policy optimization (TRPO) to reduce the FEC and emissions of both HDV and CAV.[22] While the above-mentioned RL-based controls have improved traffic smoothness by focusing on the actions of vehicles or traffic lights, the effect of combining AVs and traffic signals on FEC has not been fully investigated.[20-22]

In this study, a novel eco-driving strategy was proposed by introducing a specific percentage of AV into the traffic flow of HDV, in collaboration with smart traffic light signals to reduce the idling time of vehicles and improve the traffic smoothness in a scalable traffic network with user-defined horizontal and vertical roads for intersections. A model-free RL algorithm was developed to control the overall speed of all vehicles, resulting in a continuous traffic flow and reduction of the FEC of the vehicles in the network.

2. Method

The proposed RL algorithm determines the optimal actions of multiple agents including AV and traffic lights in a dynamic traffic network with HDV to minimize the FEC rates of all vehicles. The traffic network and motion of all vehicles are simulated using the SUMO package.[23] The RL algorithm is implemented using Python and integrated into SUMO for simulation.

The selected traffic grid environment is inspired by the grid-like layout of Manhattan City.[24] Figures 1 and 2 display an open street map of the Manhattan traffic grid structure and its visualization in the SUMO environment, respectively.

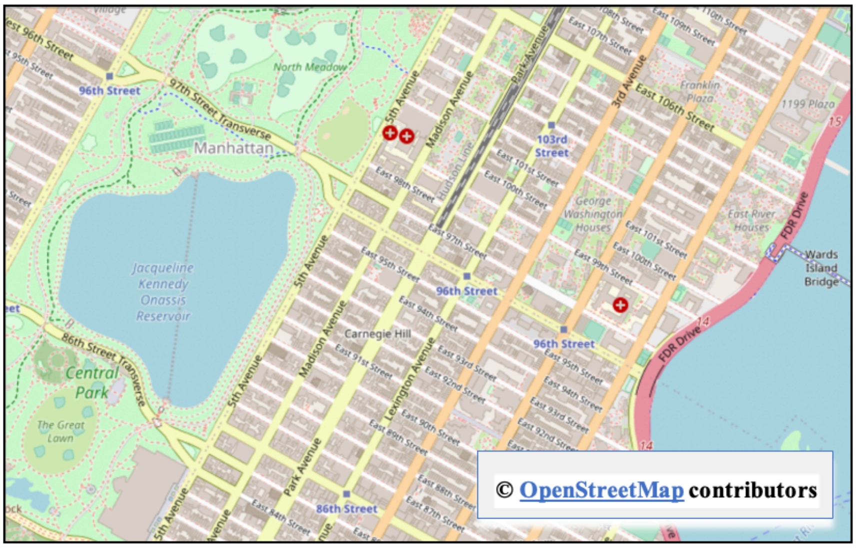

Figure 1. Open street map of traffic grid structure in Manhattan City.

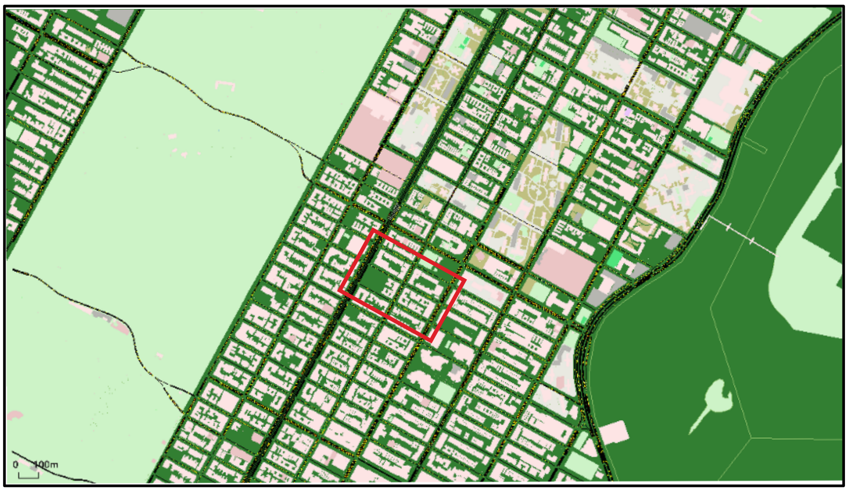

Figure 2. The grid structure of Manhattan City, simulated in the SUMO environment, is represented in the highlighted red color region. The selected traffic network serves as the basis for our research, examining the role of AV combined with HDV in minimizing the FEC rates of all vehicles in the traffic network.

Environment Setup in SUMO

The traffic network is configured within an environment featuring N horizontal and N vertical straight roads, each equipped with two lanes and extending for a length of 1 kilometer (km). There are 4N edge points, each assigned a unique number. At each edge point, a traffic flow of 300 vehicles per hour has been selected to enter the traffic system, aligning with the range of traffic flow defined by the Federal Highway Administration for signalized intersections in the United States.[25] Each vehicle has a departure speed of 30 m/s (67.1 miles/hour) and SUMO vehicle parameters dictate a minimum gap of 2.5 meters between two vehicles. All vehicles will continue straight in their original direction of travel and exit the simulation environment. To ensure safety during peak traffic time, turn prohibitions are considered in this study according to the Federal Highway Administration in the United States.[25]

According to a recent study, AV account for 10% of all vehicles on the roads.[26] Accordingly, this study considers different penetration rates for AV (0%, 5%, 10%, and 20%) to assess their impact on traffic control. An RL controller is used to control RL agents, such as AV and traffic lights, with commands issued by policy at each time step. The speed and acceleration of AV are determined with an RL controller, while the motion of HDV is controlled by an embedded "sim car-following" controller in SUMO simulation. All vehicles are homogeneous with respect to their mass, size, and economic models.

At each intersection of two roads, 4-way traffic lights are defined as actuated agents with a controllable period for red, green, and yellow lights. With the setup of N vertical and N horizontal roads in a network, there are a total of $4N^2$ traffic lights.

In this study, we focus on a 3x3 traffic network, assuming uniform road lengths in all directions to facilitate simulation. Figure 3 illustrates a network with N=3 and the arrangement of 4 traffic lights at an intersection. It's important to highlight that the framework is adaptable to larger-sized traffic networks, provided there are sufficient computational resources.

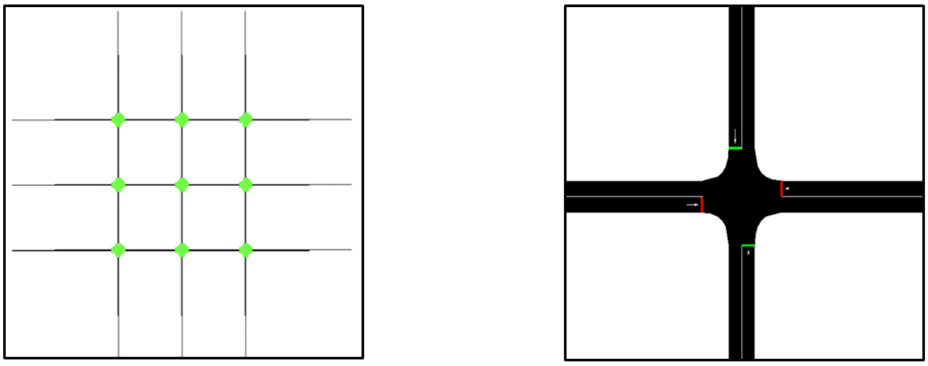

Figure 3. (a) 3x3 Traffic light grid environment (b). 4-way single signalized intersection.

Reinforcement Learning

A decentralized partially observable Markov Decision Process (De-POMDP) is adopted to coordinate the actions of agents, including traffic lights and AV in the traffic network. When vehicles move in the same direction, HDV are observable to an autonomous vehicle if the distance between a human-driven vehicle and an autonomous vehicle is less than or equal to 25 meters in the same lane. Each traffic light agent also observes the two nearest vehicles and has their information related to speed, distance to the intersection, and edge number. The position, speed, and acceleration of AV, as well as cycles and status of traffic lights, are shared among all AV and traffic lights.

The state, action space, policy, and reward function of the RL are defined as follows.

State Space (s): For each vehicle agent, its state, $s \coloneqq (v_i\,, d_i, e_i)_{i=1:M} \in R^{3XM}$, where M=3,600 denotes the maximum number of vehicles in the selected traffic system. This number is calculated by considering 300 vehicles entering the system at 4N edge points within one hour, with N=3, assuming the worst case: no vehicles leave the simulated traffic network within an hour. Here, v_i represents the speed of the i^{th} vehicle, d_i denotes the distance of the i^{th} vehicle to the nearest intersection in its driving direction, e_i indicates the edge number which the i^{th} vehicle enters the traffic network. The edge number signifies the traffic flow direction of each vehicle, assuming no turns are allowed.

The state of each traffic light agent includes the time of the light's last change, the traffic flow direction controlled by the light (0 indicates passing with a green light, and 1 indicates stopping with a red light status), and the states of other traffic lights in the same traffic flow direction. At an intersection, if the top-bottom traffic lights have a status of "0", the left-right traffic lights must have a status of "1", and vice versa. When the status of a traffic light is green, it will change to yellow for 3s before switching to red status due to safety purposes.

Action Space ($\boldsymbol{a}$): There are $4N^2$ traffic lights in the network and each traffic light have two discrete actions: 1 (indicating the traffic light switches) and -1 (indicating no action taken), as defined in equation (1):

$$\left[a = \begin{cases} 1 & t\text{raffic light switches} \\ -1 & \text{no action taken} \end{cases} .\right] \qquad (1)$$

The action space for an autonomous vehicle is its acceleration, which ranges between [-1, 1] and is determined by an RL controller in FLOW package. For HDV, the action space for acceleration values is chosen within the range [-4.5, 2.6], as defined by SUMO.

Policy: An RL algorithm called PPO is used to train policies for tasks involving decision-making in environments with either continuous or discrete action spaces. Policies are optimized using the policy gradient method to maximize the expected cumulative reward. The choice of a PPO-based RL algorithm for deployment in this study stems from its superior computational efficiency and stability compared to other algorithms. Specifically, RLlib within the Flow package is integrated into SUMO for simulation.[27-29] A stochastic policy $\pi_{\varphi}: s \times a \rightarrow \mathbb{R}_{+}$ is a maping from state, s, and action a of all agents parameterized by φ to a non-negative real number. It can be defined by (2) as a probability distribution over actions of each state:

$$\left[\pi_{\varphi} = P(a|s;\ \varphi) = \frac{e^{f_{\varphi}(s,a)}}{\sum_{a' \epsilon \mathcal{A}}^{U} e^{f_{\varphi}(s,a')}}\right] \tag{2}$$

where, $f_{\varphi}(s, a') = \varphi^{\intercal}\ \mathcal{B}(s, a)$; $\varphi = (\varphi_{a1}, \dots \varphi_{aU}) \in R^{U}$; $\varphi^{\intercal}$ is transpose of parameter vector φ; and $\mathcal{B}(s, a)$ represents transitions among states given an action; and U represents the complete action space.

The average reward received by an agent when it follows a PPO policy at each time step is referred to as the average policy reward. The average policy reward, also defined as expected return of policy, $\eta(\pi_\varphi)$ for the entire trajectory τ at time step *t*, can be expressed as equation (3),

$$\left[\eta(\pi_\varphi) = \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t . r(s_t, a_t)\right],\right] \tag{3}$$

where, τ represents the entire trajectory of states and actions. The parameter $0 < \gamma \leq 1$, represents a discount factor, and γ^t gets smaller as time $t \to \infty$ with $\gamma < 1$. The rewards function, $r(s_t, a_t)$, determines rewards given the state and action of an agent at time t. The optimal policy parameter φ^* is reached by maximizing the expected cumulative return obtained by an agent, as described in equation (4),

$$[\varphi^* := argmax_{\ \varphi}\, \eta(\pi_\varphi).] \tag{4}$$

The policy loss is defined based on the $q_t(\varphi)$, a ratio of new policy $\pi_\varphi(a_t|s_t)$ and the previous policy $\pi_{\varphi_g}(a_t|s_t)$ as equation (5):

$$\left[Policy\ Loss = \mathbb{E}_\tau\left[q_t(\varphi)\hat{A}_t - \beta KL\left[\pi_{\varphi_g}(.\,|s_t), \pi_\varphi(.\,|s_t)\right]\right],\right] \tag{5}$$

where, β is hyperparameter to control the strength of regularization of $KL[\pi_{\varphi_g}(.\,|s_t), \pi_\varphi(.\,|s_t)]$, which represents the Kullback-Leibler (KL) divergence between two conditional probability distributions over actions given a state s_t. If

$\mathbb{E}_{\tau}\left[KL\left[\pi_{\varphi_g}(.\,|s_t), \pi_{\varphi}(.\,|s_t)\right]\right] < \frac{KL\ target\ value}{1.5}$, it indicates new policy doesnot diverged significantly from the old policy, so β needs to be reduced by $1/2$. If $\mathbb{E}_{\tau}\left[KL\left[\pi_{\varphi_g}(.\,|s_t), \pi_{\varphi}(.\,|s_t)\right]\right] > (KL\ target\ value) \times 1.5$, it means there is too much change in policy through update, so β needs to be increased by multiplying with 2. The $KL\ target\ value$ is defined by users and a reference value is given in the Results section. The advantage estimate function $\hat{A}_t$, representing accumulated future rewards, can be defined as equation (6),

$$\left[\hat{A}_t = \delta_t + \sum_{d=1}^{T-t+1} (\gamma\lambda)^d \delta_{t+d}\,,\right] \tag{6}$$

$$\left[\delta_t = r_t + \gamma\, V_{(s_{t+1})} - V_{(s_t)},\right] \tag{7}$$

where t represents time steps from $[0, T]$, and T represents the range of prediction. The parameter λ impacts weights of potential rewards in the advantage estimation function $\hat{A}_t$. When λ=1, $\hat{A}_t$ increases by adding more future rewards, resulting in high variance and less bias. When λ=0, no future rewards are considered. Policy $\pi_{\varphi_{g+1}}$ is updated with φ_{g+1} according to (8):

$$\left[\varphi_{g+1} = argmax_{\varphi} \frac{1}{|\mathcal{H}_g|T} \sum_{\mathcal{T}=\mathcal{H}_g} \sum_{t=0}^{T} min\left(q_t(\varphi) A^{\pi_{\varphi g}}(s_t, a_t) - \beta_g KL[\pi_{\varphi_g}(.\,|s_t), \pi_{\varphi}(.\,|s_t)]\right),\right] \tag{8}$$

where $\mathcal{H}_g = [\mathcal{T}_i]$ is a set of trajectories for iteration g.

In the RL algorithm, the value function $V_{(s_t)}$ estimates the expected cumulative reward starting from a specific state s_t, that the agent can attain from that state onwards. A value function loss (VF Loss) is defined as a squared-error loss between predicted and target value function (9):

$$\left[VF\ Loss = (V_{\emptyset_g}(s_t) - V_t^{targ})^2,\right] \tag{9}$$

where, $V_{\varphi}(s_t)$, is an output from a neural network parameterized by $\emptyset$ with the state s_t as input; and V_t^{targ} is the target value function at time step *t* can be defined as $V_t^{targ} = r_t + \gamma V_{(s_{t+1})}$, the range of $V_t^{targ} \in [-1, 1]$. Parameters of the network $\emptyset_{g+1}$ can be updated according to (10):

$$\left[\emptyset_{g+1} = argmin_{\emptyset} \frac{1}{|\mathcal{H}_g|T} \sum_{\mathcal{T}=\mathcal{H}_g} \sum_{t=0}^{T} (V_{\emptyset_g}(s_t) - V_t^{targ})^2\right]. \tag{10}$$

In RL, the entropy function refers to the level of uncertainty in the policy distribution. It is used to encourage exploration by selecting different possible actions in a specific state and to prevent premature convergence to suboptimal policies. The entropy function of the PPO algorithm is defined based on the probability of taking actions, $\pi(a|s)$, given a state s under the policy in equation (11):

$$\left[Entropy = -\sum_{a} \pi(a|s) \log \pi\ (a|s).\right] \tag{11}$$

A smaller entropy indicates a better performance of the PPO algorithm. The pseudo code for the PPO algorithm is shown as follows.

Algorithm 1. PPO

Input: Initial policy and value function parameters $(\varphi_0, \emptyset_0)$

for iteration g=0,1, 2....do

Run policy $\pi_g = \pi(\varphi_g)$ in environment for time steps T to collect a set of trajectories $\mathcal{H}_g = [\mathcal{T}_i]$.

Compute rewards-to-go $\hat{r}_t$.

Compute advantage estimates $\hat{A}_t$ based on the current value function $V_{\emptyset_g}$.

Find optimal policy ${\varphi_g}^*$ to find average policy reward.

Update policy $\pi_{\varphi_{g+1}}$ with φ_{g+1} using equation (8).

Fit value function $V_{\emptyset_g}$ with $\emptyset_{g+1}$ using equation (10).

end for

Reward (r): Two reward functions have been designated: one to minimize total traffic delay, T_d, and another to minimize FEC rates at time step t. These reward functions were used to train each traffic light and autonomous vehicle. The reward functions are given in equations (12) and (13):

$$\left[r_1(t) = -\frac{1}{4N^2}\ T_d,\right] \tag{12}$$

$$\left[r_2(t) = -\frac{1}{\mathrm{M}}Fc(t).\right] \tag{13}$$

Since rewards are negative, the closer a reward to zero means smaller total delay and FEC rates of all vehicles in the traffic flow. The T_d is defined by (14):

$$\left[T_d = \max\left(\frac{\sqrt{\sum_i^M \left(v_{ds_i}\right)^2} - \sqrt{\sum_i^M \left(v_{ds_i} - v_i\right)^2}}{\sqrt{\sum_i^M \left(v_{ds_i}\right)^2}}, 0\right),\right] \tag{14}$$

where v_{ds} is the speed limit on the road and v_i is the velocity of each vehicle.

Fuel Energy Consumption Rate Model

The function. $Fc\ (t)$, denotes the FEC rate with a unit in Litter/second (L/s) , which is described as follows according to a previous study,[5]

$$\left[Fc\ (t) = \begin{cases} \alpha_0 + \alpha_1 P_t + \alpha_2 {P_t}^2, & \forall P_t \geq 0 \\ \alpha_0, & \forall P_t < 0 \end{cases}'\right] \tag{15}$$

$$\left[P_t = \left(\frac{R_t + 1.04 m a_t}{3600\, \eta_d}\right) \cdot v_t,\right] \tag{16}$$

$$\left[R_t = \frac{\rho}{25.92} C_d C_h A_f {v_t}^2 + 9.8066 m \frac{C_r}{1000}(c_1 v_t + c_2) + 9.8066 m G_t,\right] \tag{17}$$

Here,

- P_t : Power exerted at time t (Kilowatt, KW),
- a_t: Acceleration of vehicle (m/s^2),
- v_t: Velocity of vehicle (m/s),
- R_t : Resistance force (N).

Other constant parameters in the model have been defined in Table 1.

Table 1. FEC rate model parameters

Symbol	FEC rate Parameters	Value
α_0	Vehicle model constant	0.00000002
α_1	Vehicle model constant	0.0000001
α_2	Vehicle model constant	0.000001
m	Vehicle mass	1200 Kg
η_d	Derive line efficiency	0.92
ρ	Density of air at sea level at a temperature of 59°F	1.2256 Kg/m^3
C_d	Drag coefficient	0.28
C_h	Correction factor for altitude	0.97
A_f	Frontal area	2.6 m^2
C_r	Rolling coefficient	1.75
c_1	Rolling resistance parameter	0.0328
c_2	Rolling resistance parameter	4.575
G_t	Roadway grade	0.04

Computational Framework

Two publicly available software packages, Flow and SUMO, are adopted in this study. Flow is a traffic control benchmarking framework developed in Python and integrates RL algorithms into different traffic control scenarios.[19] The SUMO simulator handles large-scale traffic networks based on physical-world data. Integration of SUMO and Flow package and implementation of the RL algorithm are shown in Figure 4.

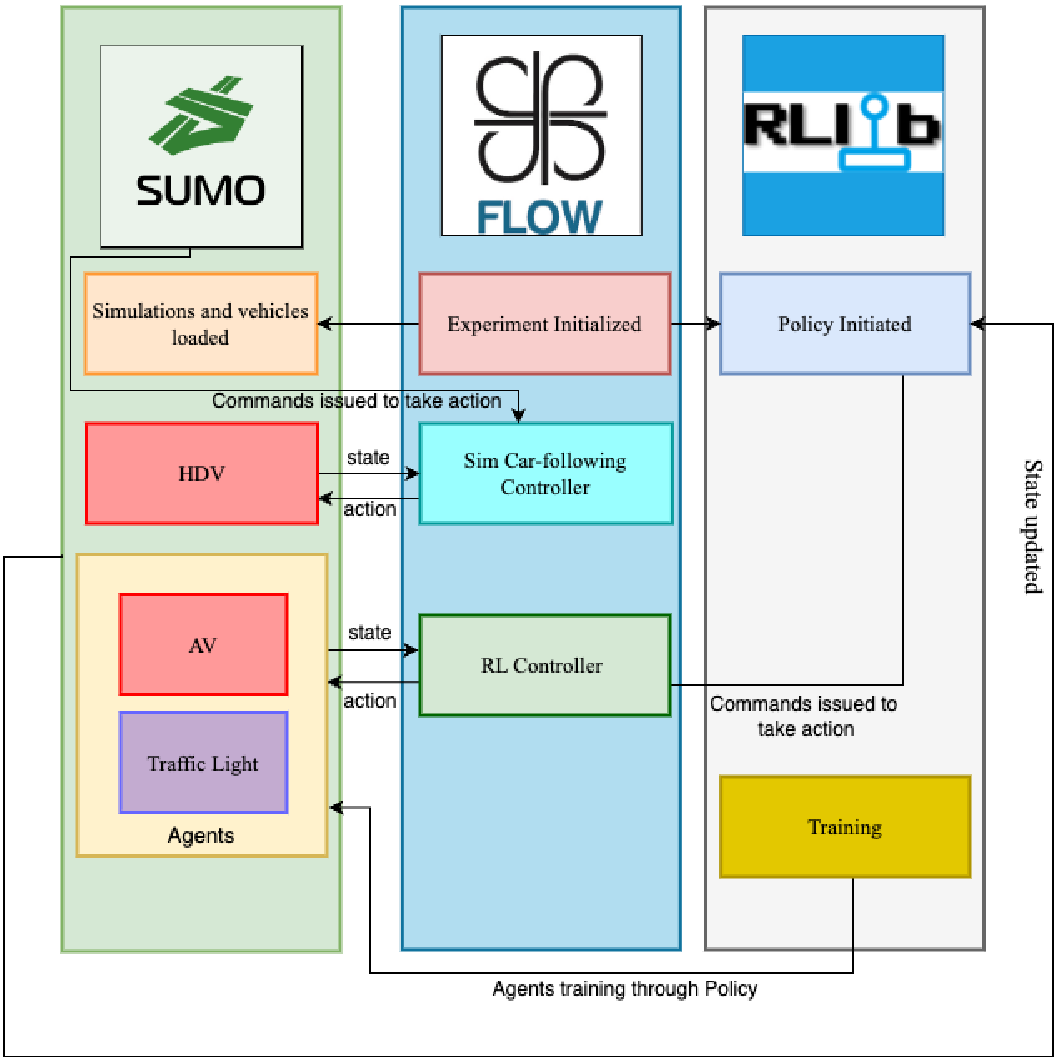

Figure 4: Process diagram to describe RL training process and interactions between SUMO, Flow, and RLlib library. RL and Sim car following controllers used to control the AV and HDV, respectively. Sim

car-following controller actions are entirely defined by the simulator, whereas RL-Controller performs actions by following commands from the policy in RLlib.

Results

The RL algorithm was applied to regulate the traffic flow in the selected traffic network with 4 different penetration rates of AV, 0%, 5%, 10%, and 20%.

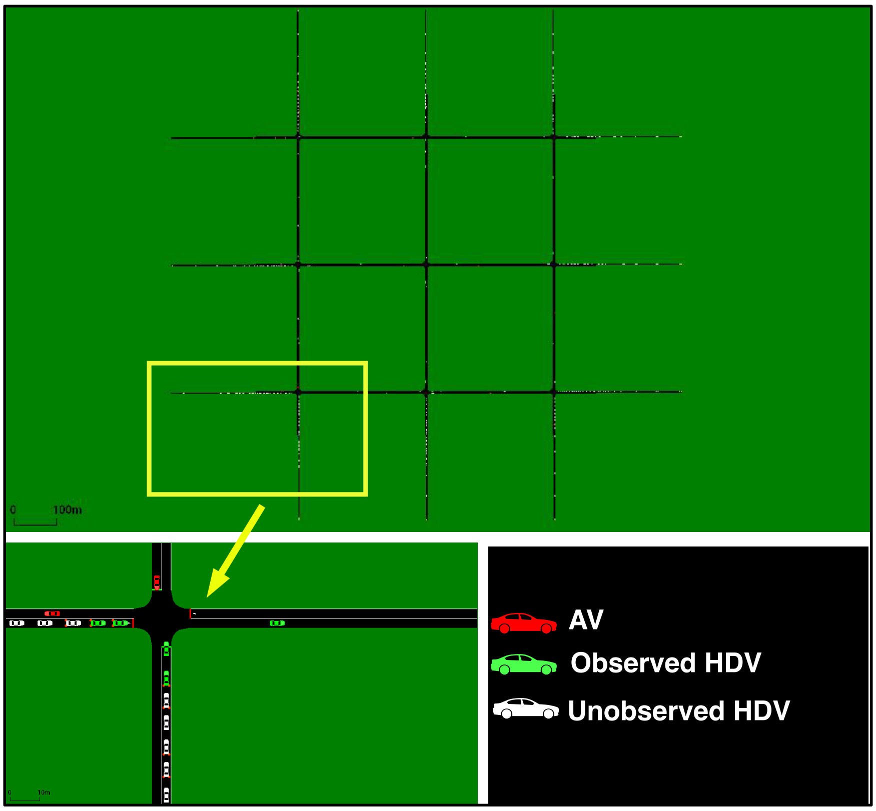

Figure 5. Illustration of the traffic network with 3 vertical and 3 horizontal roads in SUMO simulator. An overview of all AV (red vehicles), observable HDV (green vehicles), and unobservable HDV (white

vehicles) in the traffic network. (Bottom Left) A close view of traffic flow between intersections within the yellow box at the lower left part of the traffic network.

Training was conducted on a machine with 4 Intel® Core™ i5-6600 CPU @ 3.30GHz.

The Hyperparameters used in the RL algorithms are listed in Table 2.

Table 2. RL algorithm hyperparameters

Hyperparameters	**Value**
Learning rate	5e-5
Training batch size	1500
SGD minimum batch size	128
Number of SGD iterations	5
Training Iterations	500
Parallel workers	10
Horizon steps	150
Discount factor (γ)	0.999
GAE value (λ)	0.5
KL Target value	0.02
Target value function	0.01
Fixed KL β	3

SGD: Stochastic Gradient Descent; GAE: Generalized Advantage Estimation; KL: Kullback-Leibler.

The results of this study have been divided into four categories:

(a) Rewards on total delay at different penetration levels;

(b) Rewards on FEC rates at different penetration levels;

(c) Performance of PPO policy at different penetration levels;

(d) Comparison of the selected 3x3 traffic network with other networks.

Rewards on Total Delay at Different Penetration Levels

Various penetration rates of AV in the traffic flow are examined to optimize traffic flow considering information sharing on traffic lights and AV.

Figure 6 shows that traffic flow containing 100% HDV (i.e., 0% AV) has the worst total delay rewards in the long term compared to 5%, 10%, and 20% penetration rates of AV. Penetration of AVs at 5%, 10%, and 20% results in convergence of rewards on total delay. The 20% AV penetration rates show a more complicated learning process due to the priority of safety over the optimization of traffic flow speed and FEC rate at an early stage of the learning process. Once AV become familiar with the traffic flow patterns during the training period, the PPO algorithm improves the rewards on total delays due to good prediction of other vehicles' behavior. The 10% penetration rate of AV indicates fluctuations as well during the training period and it could be due to fewer interactions between AV and HDV on the road.

Table 3 shows the average delay for different penetration rates of AV in the selected traffic grid network. At a 10% pentation rate, the average reward was achieved at the least time step of 110K as compared to other penetration rates.

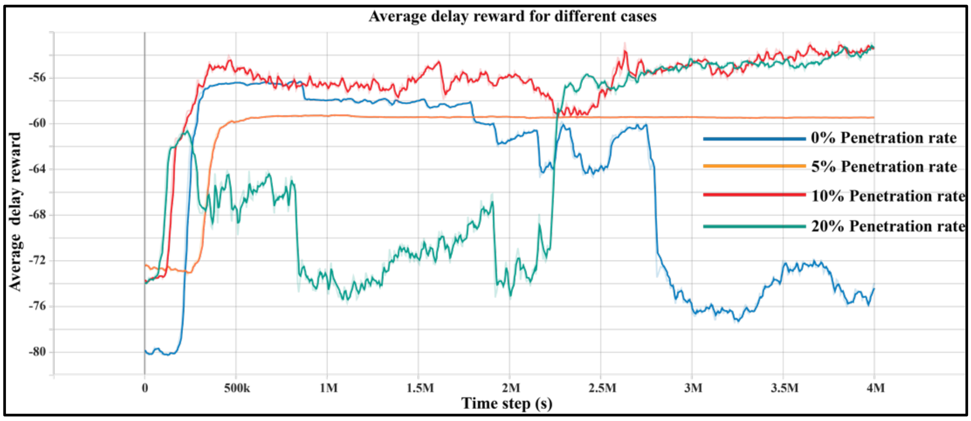

Figure 6. Behavior of average rewards of total delay with respect to time steps for different AV penetration rates.

Table 3. Convergence time and steady state average rewards on total delay obtained with different pentation rates.

Penetration rate	Approximate starting time steps of convergence	Average rewards on total delay at the last time step
0%	No exact convergence observed	-73.94
5%	302K	-59.79
10%	110K	-53.34
20%	2.24M	-53.51

FEC Rates at Different Penetration Levels

The FEC rate of a small-engine vehicle usually falls within the range of 0.05-0.10 L/s with an average driving velocity. The reward on FEC rate can reach zero when the vehicle achieves low levels of FEC at a minimum varying speed and other performance parameters. Results of the average rewards on FEC rate obtained from different penetration levels are presented in Figure 7 and Table 4. The penetration of AV shows better performance in reaching larger rewards on FEC rates, while the pure HDV case illustrates the worst scenario with a reward on FEC rate of about -1,000. Interestingly, the 10% pentation rate regulates the FEC rate faster in the simulation compared to the results obtained from 5% and 20% penetration of AV. The time step for the convergence of the FEC rate and the steady state average rewards on the FEC rate with 4 different penetration rates are presented in Table 4.

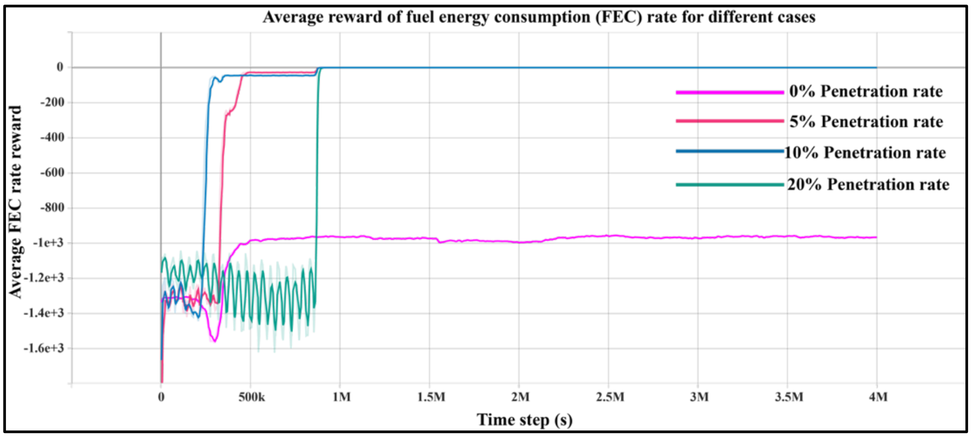

Figure 7. Behavior of average rewards on FEC rates with respect to time steps, considering 4 different penetration rates of AV in the traffic flow.

Table 4. FEC rate results at different pentation rates.

Penetration rate	Starting time step of convergence	Average rewards on FEC rate at the last time step
0%	302K	-964.3
5%	316k	0.071

10%	216k	0.010
20%	862K	0.081

Performance of PPO policy

To assess the effectiveness of the PPO policy in optimizing the FEC rate, the average policy reward and average environment time, policy loss, entropy, and value function loss have been evaluated with different pentation rates of AV. Figure 8 shows that average policy rewards with penetration rates 5%, 10%, and 20% of AV converge to zero while HDV only case has the worst reward with a value of -107.2. With the 10 % penetration rate, the average policy rewards start to converge about 300K steps, faster than 0%, 5%, and 20% penetration rates.

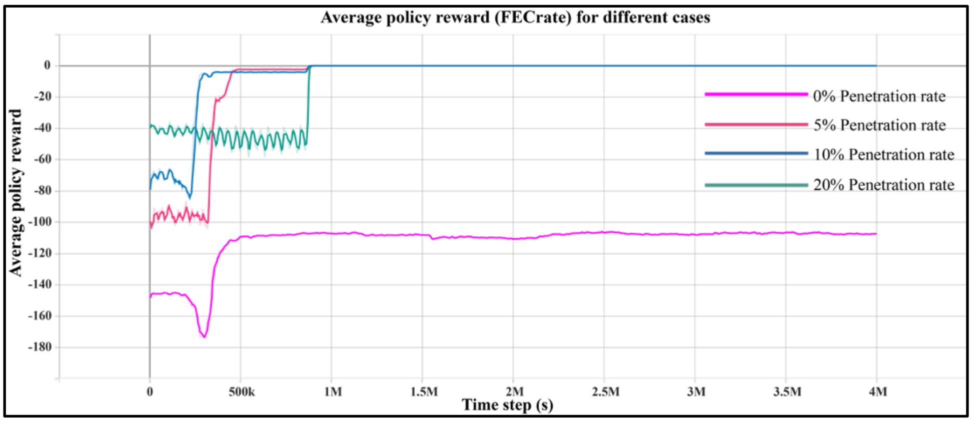

Figure 8. Behavior of average policy rewards with respect to time steps for 4 different penetration rates of AV.

The time for an agent to stay in a specific state before applying an action is considered an environment waiting time. The highest average environment time is observed for the 20% penetration rates at a time step of about 850K as compared to others, as shown in Figure 9. The average environment waiting time of 5% case is slightly higher than that of the 10% AV penetration rate by the end of training, but its highest peak is observed to be higher than the 10% penetration rates during training at 800K steps.

The entropy behavior for PPO is shown in Figure 10, with high values observed at a 0% penetration rate. A minimum value of entropy was observed at a 10% penetration rate by the end of training, indicating fewer uncertainties in policy distribution as compared to the 5% and 20% penetration rates.

The total loss, which is a combination of policy loss and value function loss, is depicted in Figure 11. At a 10% penetration rate, the minimum total loss is observed compared to other cases. All these performance indices show that penetration of AV can improve the rewards on FEC rates compared with 100% HDV traffic flow.

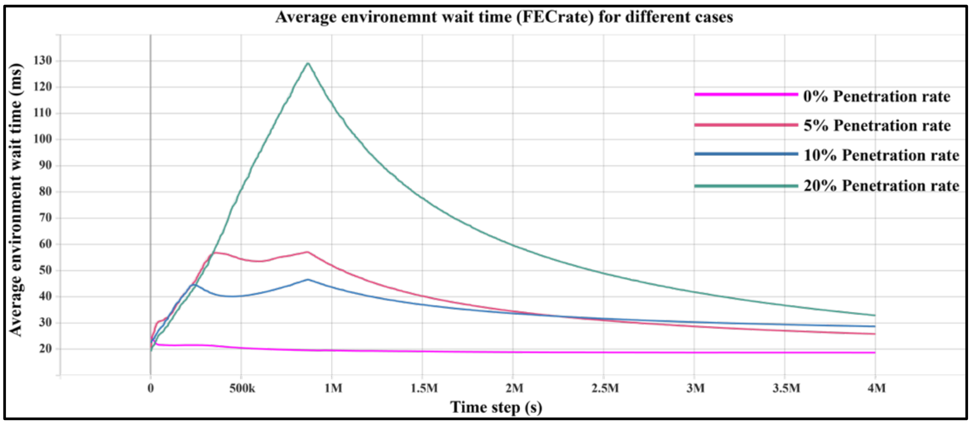

Figure 9. Average environment waiting time with respect to time steps for 4 different penetration rates of AV.

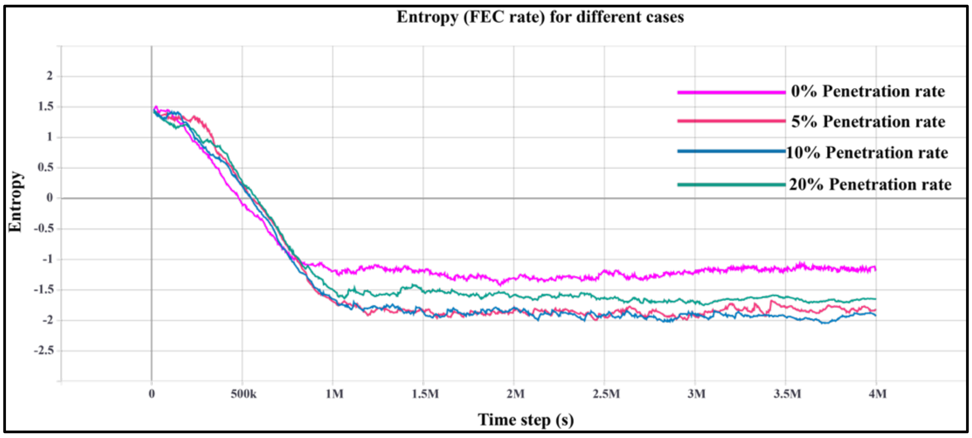

Figure 10. Entropy of the policy to optimize FEC rates for 4 different penetration rates of AV.

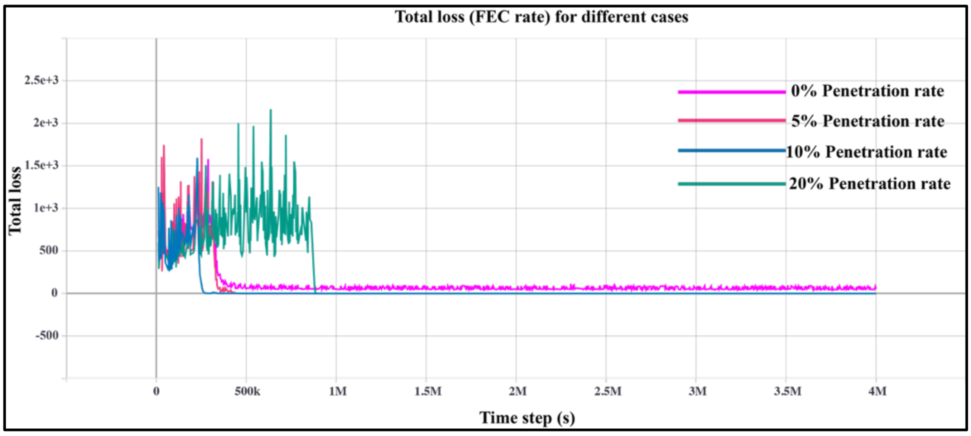

Figure 11. Total loss including policy value function loss with respect to time steps for 4 penetration rates of AV.

Table 5 shows the values of five measurements of policy to optimize FEC rate for 4 different penetration rates of AV.

Table 5. FEC rate results at different pentation rates

Measurements	**Penetration Rate**			
	0%	**5%**	**10%**	**20%**
Average policy reward	-107.2	0.059	0.057	0.069
Average environment Wait time (ms)	18.7	25.81	28.69	32.87
Policy loss	6.724e-3	6.07e-3	9.225e-3	6.801e-3
Entropy	-1.158	-1.829	-1.924	-1.65
Value function loss	52.66	7.05e-7	6.268e-7	1.622e-6

Comparison with Other Traffic Grid Environments

With a 10% penetration rate of AV, four traffic environments including 1x1, 1x2, 2x2, and 3x3 traffic grids were simulated with the proposed PPO algorithm. Figures 12 to 15 show the behavior of the average rewards on FEC rates with respect to time steps for each simulated environment, respectively. Table 6 presents the convergence of average rewards on FEC rates and the convergence time for each simulated environment. Specifically, the average FEC reward in the 3X3 traffic grid converged at about 216K steps, which was less than results obtained from other environments.

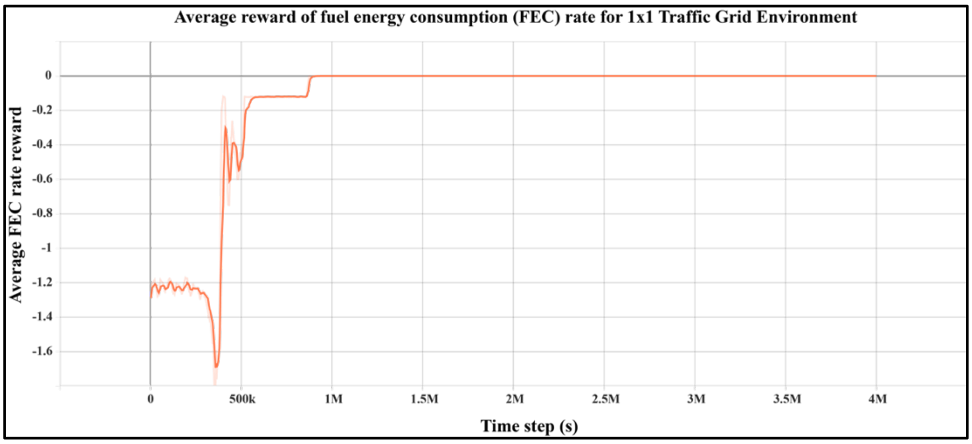

Figure 12. Behavior of average rewards on FEC rate with respect to time steps for a 1x1 traffic grid.

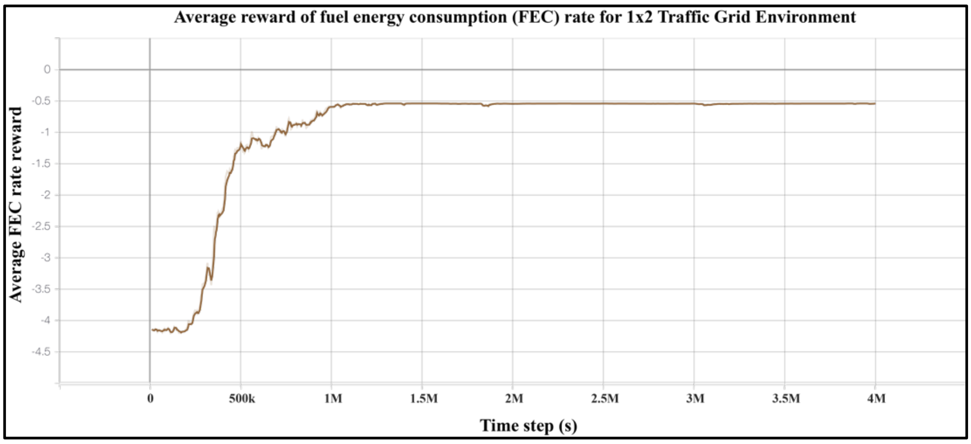

Figure 13. Behavior of average rewards on FEC rate with respect to time steps for a 1x2 traffic grid.

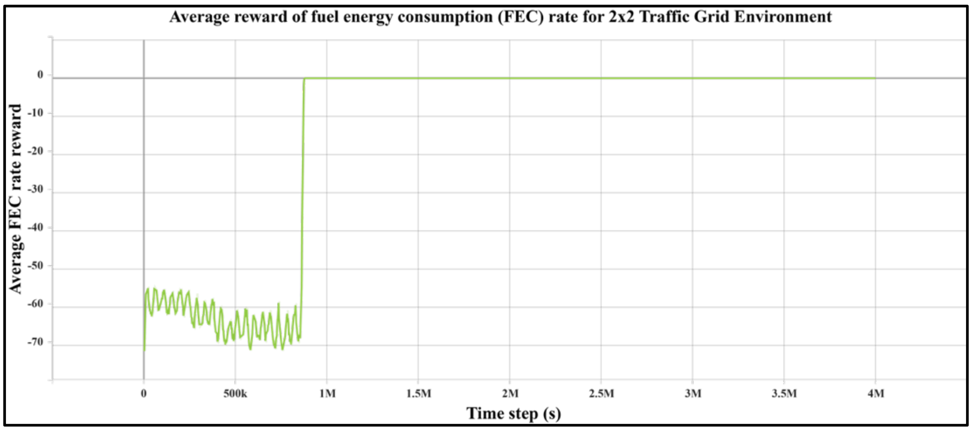

Figure 14. Behavior of average rewards on FEC rate with respect to time steps for a 2x2 traffic grid.

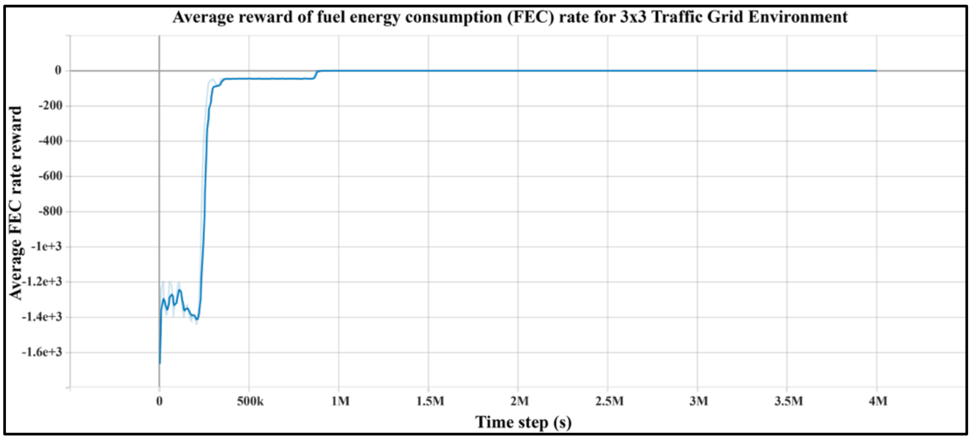

Figure 15. Behavior of average rewards on FEC rate with respect to time steps for a 3x3 traffic grid.

Table 6. Performance of the PPO algorithm for rewards on FEC rates for different traffic grid networks.

Traffic Grid	Converging Time steps	Average rewards on FEC rates at convergence
1x1	550K	-0.1
1x2	944K	-0.7231
2x2	854K	-69.4
3x3	216K	-20.1

3. Discussion

As more cars run on fuel like gasoline, resulting in air pollutants, the demand for eco-driving strategies is highly.[30] [31] In this study, we employed an RL algorithm, PPO, to investigate the impact of introducing AV to the traffic flow of HDV on reducing traffic delay and minimizing fuel energy consumption rates. This involved introducing a specific penetration rate of AV into a continuous traffic flow, coordinated with traffic light signals, within a large 3x3 traffic grid system. The Flow computational package, developed in Python, was utilized to integrate the publicly available microscopic traffic simulator, SUMO, and the RL library 'RLlib'. In a previous study,[32] a comparison between different types of action spaces for different algorithms has been presented. Algorithms such as Q-Learning, DQN, DDPG, etc. are considered reliable for specific types of action spaces—either continuous or discrete. For environments that feature both continuous and discrete action spaces, PPO-based RL algorithms are feasible due to their computational and sample efficiency. So, the PPO-based RL approach is used in this research work to train agents in the selected traffic environment.

The environmental setup consists of a traffic network with 3 horizontal and 3 vertical roads in the SUMO simulator. Different percentages of AV (0%, 5%, 10%, and 20%) were introduced in this study to control the speed of HDV in the network. The average rewards on total delay and FEC rates were computed in this research work with different penetration rates. The penetration of AV illustrated better average rewards on both total delay and FEC rates. The 20% AV penetration initially results in more delays due to their prioritization of safety over speed and efficiency. Specifically, a 10% penetration rate in AV combined with HDV showed significant results for minimization of FEC rate and total delay. The rewards on total delay for the 10% penetration rate case converged at a minimum value of -53.34 at the least time steps of 110K in comparison with other cases. At a 0% penetration rate, an average reward on FEC rates of -964.3 was obtained by the end of training. For all other cases, the rewards on FEC rates approached zero by the end of training. To assess the performance of the PPO policy in training agents to minimize the FEC rates, results for average policy reward, entropy, value function loss, and mean environment time were obtained at various penetration rates. A 10% penetration rate demonstrated better performance compared to 0%, 5%, and 20% rates. A comparison of four traffic light grids (1x1, 1x2, 2x2, and 3x3) was performed at a penetration level of 10% in terms of the FEC rate. The results indicated that the average rewards on FEC rates converged in a shorter time for a 3x3 traffic network as compared to other configurations. We are well aware that there are limitations in this study. PPO-based RL algorithms need to be precisely tuned to achieve the most effective learning results because they are hyperparameter-sensitive. The consideration of lane change was not incorporated into this research; lane-changing behavior can be discussed by introducing a lane change controller in the future. All HDV are assumed to have the same economic model, while there are heavy-duty vehicles, buses, cycles, and passenger vehicles have different economic models. To address this limitation, we can find an economical model for each type of vehicle, determine the percentage of each vehicle type based on the public traffic flow dataset, and integrate this information into the energy calculation in our future research. We assume ideal communication without any delay or failure in controlling AV in this study. In the future, we can apply RL algorithms to scaled-down AV to examine the impact of communication delays.

A comparison of the proposed eco-driving strategy is performed with prior related research as shown in Table 7, suggesting the effectiveness of the proposed PPO algorithm for an eco-driving strategy.

Table 7. Comparison with prior research work.

Reference	Year	Vehicle Type	Algorithm	Action Space Type	Traffic Grid Network	Objective	Fuel Consumption (L) per Vehicle
[17]	2018	CV	Q-Learning	Discrete	Cases-1 single intersection (1x1) Case-2: a 2-way road network	To minimize CO2 emissions and optimize traffic performance	N/A
[18]	2020	CAV	DQN	Discrete	1x1	Optimizing acceleration/deceleration of CAV to minimize fuel consumption	0.0691
[20]	2020	CAV	DDPG	Continuous	Circular Network with signalized interactions	To enhance travel efficiency, reduce fuel consumption, and ensure safety	0.015 (at 100% CAV)
[21]	2021	CAV	DDPG +DQN	Discrete & Continuous	1x5	To minimize fuel consumption, ensure reasonable travel times,	0.12005 (for HDQPG)

						and execute lane changes strategically to avoid congested lanes	
[22]	2019	CAV and HDV	TRPO	Continuous	1x1	Percentage of AV in the traffic flow to minimize fuel consumption, emissions, and improvement in travel speed	0.0954 (at 100% CAV)
This work	2024	AV and HDV	PPO	Continuous & Discrete	3x3	Collaboration of traffic lights signals and percentage of AV in the traffic flow to minimize fuel consumption and total delay	0.010 (at 10% AV)

4. Conclusions

Eco-driving positively impacts human health by reducing pollution resulting from vehicle fuel consumption and emissions. This study explores a hybrid traffic network that combines autonomous vehicles and human-driven vehicles through the coordination of traffic light signals to manage a large traffic flow. The approach addresses eco-driving challenges, including real-time traffic data collection and the intricate nature of the traffic network, which currently lacks a comprehensive mathematical model. The research employs model-free PPO-based reinforcement learning algorithms to analyze the fuel energy consumption rates of vehicles. It focuses on minimizing fuel energy consumption by introducing specific penetration rates of AV (0%, 5%, 10%, and 20%) in a 3x3 traffic grid system, utilizing the Flow compactional package to integrate the SUMO simulator and RLlib. The study results indicate that a 10% penetration rate of AV alongside HDV yielded significant reductions in both fuel consumption and total delay of traffic.

Acknowledgement

This research was partially supported by the National Science Foundation (2051113 to YFJ) and the Department of Transportation (TranSET Program 21ITS034 and 21UTSA049 to YFJ).

Author's Contribution

Conceptualization, U.J., M.F. M.X, C.D., and Y.J.; Data Curation, U.J., M.F., and Y.J..; Methodology, U.J. M.F. and Y.J.; Simulations, U.J., A.C., and M.M.; Supervision, Y.J.; Writing-original Draft Preparation, U.J., and Y.J.; Writing-Review and Editing, M.M, M.F., C.D., and M.X.; Funding, Y.J. All authors have reviewed the manuscript and provided their consent for publication.

Conflict of Interests Statement

The authors declared no conflicts of interest for this publication.

Funding

This work was supported in part by the U.S. National Science Foundation #2051113, and US Department of Transportation for Transportation Consortium of South-Central States (TranSET) with projects 21034 and 21049. The funding sources had no role in the design of the study; collection, analysis, and interpretation of data; or in the writing of the manuscript.

ORCID iD

Umar Jamil https://orcid.org/0000-0002-2346-556X

Yu-Fang Jin https://orcid.org/0000-0002-7421-527X

References

1. U.S. Energy Information Administration (EIA). Use of energy explained Energy use for transportation. Accessed October 30, 2023, https://www.eia.gov/energyexplained/use-of-energy/transportation.php#:~:text=Energy%20sources%20are%20used%20in,and%20some%20types%20of%20helicopters.
2. Pasquale C, Sacone S, Siri S, Ferrara A. Traffic control for freeway networks with sustainability-related objectives: Review and future challenges. *Annual Reviews in Control*. 2019;48:312-324.
3. Boltze M, Tuan VA. Approaches to achieve sustainability in traffic management. *Procedia engineering*. 2016;142:205-212.
4. Jamil U, Sulaiman M, Ghafoor N, Malmir M, Nawaz F, Shakoor RI. Power Harvesting towards Sustainable Energy Technology through Ambient Vibrations and Capacitive Transducers. IEEE; 2023:1-6.
5. Park S, Rakha H, Ahn K, Moran K. Virginia tech comprehensive power-based fuel consumption model (VT-CPFM): Model validation and calibration considerations. *International Journal of Transportation Science and Technology*. 2013;2(4):317-336.
6. Lee JW, Gunter G, Ramadan R, et al. Integrated Framework of Vehicle Dynamics, Instabilities, Energy Models, and Sparse Flow Smoothing Controllers. *arXiv preprint arXiv:210411267*. 2021;
7. Yao Z, Hu R, Wang Y, Jiang Y, Ran B, Chen Y. Stability analysis and the fundamental diagram for mixed connected automated and human-driven vehicles. *Physica A: Statistical Mechanics and Its Applications*. 2019;533:121931.
8. Hao P, Wei Z, Bai Z, Barth MJ. *Developing an adaptive strategy for connected eco-driving under uncertain traffic and signal conditions*. 2020.
9. Wang S, Lin X. Eco-driving control of connected and automated hybrid vehicles in mixed driving scenarios. *Applied Energy*. 2020;271:115233.
10. Zhou S, Xie M, Jin Y, Miao F, Ding C. An End-to-end Multi-task Object Detection using Embedded GPU in Autonomous Driving. 2021:122-128.

11. Wei H, Zheng G, Gayah V, Li Z. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD Explorations Newsletter*. 2021;22(2):12-18.
12. Clemmons J, Jin YF. Reinforcement Learning-Based Guidance of Autonomous Vehicles. 2023:1-6.
13. Zeynivand A, Javadpour A, Bolouki S, et al. Traffic flow control using multi-agent reinforcement learning. *Journal of Network and Computer Applications*. 2022;207:103497.
14. Li M, Cao Z, Li Z. A reinforcement learning-based vehicle platoon control strategy for reducing energy consumption in traffic oscillations. *IEEE Transactions on Neural Networks and Learning Systems*. 2021;32(12):5309-5322.
15. Haydari A, Zhang M, Chuah C-N, Ghosal D. Impact of deep rl-based traffic signal control on air quality. IEEE; 2021:1-6.
16. Stern RE, Chen Y, Churchill M, et al. Quantifying air quality benefits resulting from few autonomous vehicles stabilizing traffic. *Transportation Research Part D: Transport and Environment*. 2019;67:351-365.
17. Shi J, Qiao F, Li Q, Yu L, Hu Y. Application and evaluation of the reinforcement learning approach to eco-driving at intersections under infrastructure-to-vehicle communications. *Transportation Research Record*. 2018;2672(25):89-98.
18. Mousa SR, Ishak S, Mousa RM, Codjoe J, Elhenawy M. Deep reinforcement learning agent with varying actions strategy for solving the eco-approach and departure problem at signalized intersections. *Transportation research record*. 2020;2674(8):119-131.
19. Vinitsky E, Kreidieh A, Le Flem L, et al. Benchmarks for reinforcement learning in mixed-autonomy traffic. PMLR; 2018:399-409.
20. Zhou M, Yu Y, Qu X. Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach. *IEEE Transactions on Intelligent Transportation Systems*. 2019;21(1):433-443.
21. Guo Q, Angah O, Liu Z, Ban XJ. Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors. *Transportation Research Part C: Emerging Technologies*. 2021;124:102980.
22. Jayawardana V, Wu C. Learning eco-driving strategies at signalized intersections. IEEE; 2022:383-390.
23. Lopez PA, Behrisch M, Bieker-Walz L, et al. Microscopic traffic simulation using SUMO. IEEE; 2018:2575-2582.
24. OpenStreetMap Contributors. Manhattan City [Map]. Accessed June 22, 2023, https://www.openstreetmap.org/export#map=15/40.7867/-73.9533

25. Federal Highway Administration. Chapter 12 - Signalized Intersections: Informational Guide, August 2004 (Publication Number: FHWA-HRT-04-091). Accessed November 22, 2022, 2022. https://www.fhwa.dot.gov/publications/research/safety/04091/12.cfm
26. McKinsey and Company. Autonomous driving's future: Convenient and connected. Accessed January 20, 2023, 2023. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected
27. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. *arXiv preprint arXiv:170706347*. 2017;
28. Liang E, Liaw R, Nishihara R, et al. RLlib: Abstractions for distributed reinforcement learning. PMLR; 2018:3053-3062.
29. Wu C, Kreidieh AR, Parvate K, Vinitsky E, Bayen AM. Flow: A modular learning framework for mixed autonomy traffic. *IEEE Transactions on Robotics*. 2021;38(2):1270 - 1286.
30. Huang Y, Ng EC, Zhou JL, Surawski NC, Chan EF, Hong G. Eco-driving technology for sustainable road transport: A review. *Renewable and Sustainable Energy Reviews*. 2018;93:596-609.
31. Malmir M, Momeni H, Ramezani A. Controlling Megawatt Class WECS by ANFIS Network Trained with Modified Genetic Algorithm. IEEE; 2019:939-943.
32. Zhu J, Wu F, Zhao J. An overview of the action space for deep reinforcement learning. 2021:1-10.

Author Biographies

Umar Jamil is a Doctoral Candidate and Graduate Research Assistant in the Electrical and Computer Engineering Department at the University of Texas at San Antonio, USA. He received his MS in Mechatronics Engineering from Air University, Islamabad, Pakistan. Additionally, he graduated with another MS degree in Electrical Engineering, and a BS degree in Electrical Engineering, focusing on Power Systems, from Mirpur

University of Science and Technology, Azad Jammu & Kashmir, Pakistan. His research interests encompass intelligent transportation systems, deep learning, reinforcement learning, electrical power distribution systems, sensors and actuators, and energy harvesting systems.

Mostafa Malmir is a second-year Ph.D. student at the Electrical Engineering Department of the University of Texas at San Antonio. His research concentration is studying the advantages of different Machine Learning and Deep Learning algorithms and explore implementing them in biological applications. His current work examines novel methods of single-cell cell typing to improve cell identification for rare cell types.

Monika Filipovska is an assistant professor of Transportation and Urban Engineering in the Department of Civil and Environmental Engineering at the University of Connecticut. She received her Ph.D. and MS in Civil and Environmental Engineering, focusing on Transportation Systems Analysis and Planning, from Northwestern University. Her research interests are in the domains of transportation networks and traffic modeling, with a focus on emerging vehicles and sensing technologies and data-driven predictive analytics.

Mimi Xie is an assistant professor in Computer Science at the University of Texas at San Antonio. She received the BE and MS degrees from the College of Computer Science, Chongqing University, Chongqing, China, in 2010 and 2013, respectively, and the Ph.D.

degree in Electrical and Computer Engineering from the University of Pittsburgh in 2019. She is currently an assistant professor with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX. Her current research interests include energy-harvesting embedded Systems and AI on Edge.

Caiwen Ding received the Ph.D. degree in computer science and engineering from the Northeastern University, Boston, MA, USA, in 2019. He is currently an Assistant Professor in the Department of Computer Science and Engineering at the University of Connecticut, Storrs, CT, USA. His research interests include machine learning and deep neural network systems, computer vision, and natural language processing. Dr. Ding is the recipient of the Best Paper Award Nomination at DATE 2018 and DATE 2021.

Yu-Fang Jin received her Ph.D. degree in Electrical and Computer Engineering from the University of Central Florida, Orlando, Florida, USA, in 2004. She is currently a Full Professor in the Department of Electrical and Computer Engineering at the University of Texas at San Antonio, San Antonio, Texas, USA. Her research interests include applications of deep learning to large-scale networked systems and interpretations of deep learning algorithms.