


Article

Energy-Efficient Driving for Adaptive Traffic Signal Control Environment via Explainable Reinforcement Learning

Xia Jiang ¹, Jian Zhang ^{1,2,*} and Bo Wang ¹

¹ Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, and Jiangsu Province Collaborative Innovation Center for Technology and Application of Internet of Things, School of Transportation, Southeast University, Nanjing 210096, China; 220203381@seu.edu.cn (X.J.); aisijimewb@163.com (B.W.)

² School of Engineering, Tibet University, Lhasa 850000, China

* Correspondence: jianzhang@seu.edu.cn

Abstract: Energy-efficient driving systems can effectively reduce energy consumption during vehicle operation. Most of the existing studies focus on the driving strategies in a fixed signal timing environment, whereas the standardized Signal Phase and Timing (SPaT) data can help the vehicle make the optimal decisions. However, with the development of artificial intelligence and communication techniques, the conventional fixed timing methods are gradually replaced by adaptive traffic signal control (ATSC) approaches. The previous studies utilized SPaT information that cannot be applied directly in the environment with ATSC. Thus, a framework is proposed to implement energy-efficient driving in the ATSC environment, while the ATSC is realized by the value-based reinforcement learning algorithm. After giving the optimal control model, the framework draws upon the Markov Decision Process (MDP) to make an approximation to the optimal control problem. The state sharing mechanism allows the vehicle to obtain the state information of the traffic signal agents. The reward function in MDP considers energy consumption, traffic mobility, and driving comfort. With the support of traffic simulation software SUMO, the vehicle agent is trained by Proximal Policy Optimization (PPO) algorithm, which enables the vehicle to select actions from continuous action space. The simulation results show that the energy consumption of the controlled vehicle can be reduced by 31.73%~45.90% with a different extent of mobility sacrifice compared with the manual driving model. Besides, we developed a module based on SHapley Additive exPlanations (SHAP) to explain the decision process in each timestep of the vehicle. That can make the strategy more reliable and credible.

Keywords: energy efficient driving; electric vehicles; reinforcement learning; signalized intersection



Citation: Jiang, X.; Zhang, J.; Wang, B. Energy-Efficient Driving for Adaptive Traffic Signal Control Environment via Explainable Reinforcement Learning. *Appl. Sci.* **2022**, *12*, 5380. <https://doi.org/10.3390/app12115380>

Academic Editor: Konstantinos Gkoumas

Received: 18 April 2022

Accepted: 21 May 2022

Published: 26 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fickle driving behaviors can easily lead to increased carbon emissions and energy consumption [1], which then becomes a source of global warming and the greenhouse effect. In order to build a eco-friendly transportation system, the application of electric vehicles and advanced driving strategies is necessary [2]. In this case, energy efficient driving, also known as “eco-driving”, is an important way to achieve energy conservation and emission reduction for vehicles [3]. It is usually implemented by avoiding sudden acceleration and reducing idling time [4], whereas frequent acceleration/deceleration, slow movements, and idling are the main factors affecting vehicular energy consumption. Generally, both freeway [5,6] and signalized intersections of urban roads are the primary scenarios for eco-driving [7,8], but the intersection scenario is more challenging because of the uncertainty of the changing signal phase and the evolving road traffic. Traffic signals and vehicle queues impose spatial and temporal constraints on the movement of the vehicle needed to be controlled (i.e., the ego vehicle) [9]. However, with the support of advanced vehicle-to-infrastructure (V2I) communication and vehicle-to-vehicle (V2V) communication, the

uncertainty can be attenuated to help the vehicle make better decisions to cross the conflict area [10].

Scholars have presented many methods to realize eco-driving in the proximity of signalized intersections. More specifically, most of the existing works utilize signal phase and timing (SPaT) information so that the ego vehicle can achieve the energy-saving goal [11]. The SPaT data acquired by the ego vehicle through V2I communication includes the current phase and the phase changing time, which can guide the vehicle to adjust its acceleration to avoid a red light. Leveraging the SPaT data, the majority of previously proposed studies obtain the control law of the vehicle by formulating an optimal control problem [12–14]. More specifically, Asadi and Vahidi [15] reported a multi-objective adaptive cruise control (ACC) system to reduce idle time and fuel consumption by formulating an optimization-based control algorithm. Lin et al. [16] considered the powertrain dynamics of a CV and solved the eco-driving problem numerically by combining the multi-stage driving rule and the Dijkstra algorithm to get an approximated optimal trajectory. Wang et al. [17] introduced a connected and automated vehicle (CAV)-based method targeted for a signalized corridor scenario. The CAVs in their paper can receive SPaT and serve as leaders or followers in a string to achieve cooperative eco-driving. Mousa et al. [18] put forward a framework for CV passing through semi-actuated signalized intersections. They applied a simple algorithm to predict the signal changing time with the premise that the logic of signal timing is known, so the trajectory of the ego vehicle can be planned. Yang et al. [19] also proposed a SPaT-based trajectory planning approach, which was designed in a modular and scalable fashion. They carried out a series of comprehensive sensitivity analyses to explore the fuel-optimum configuration of the traffic phase and system penetrating rate. Moreover, Dong et al. [20] built a queue discharge prediction model and controlled the ego vehicle according to the queue estimation results. Dynamic programming (DP) and model predictive control (MPC) are used to design the general speed profile of the vehicle and compute the explicit control solution, respectively. Besides, some researchers extend the control framework to the platoon-based situations, which also utilized SPaT data to further reduce the energy consumption of the vehicles [21,22].

In addition to these optimization-based methods, some rule-based methods are also implemented, aiming at reducing the computation complexity. In general, the rule-based frameworks divided the running of CVs into several categories with regard to vehicular position and traffic signal status [23,24]. By acquiring the status of traffic lights, the controlled CVs know if it should speed up or slow down, judging whether the vehicle could reach the junction within the duration of the green light.

In fact, the works mentioned above are applicable to pre-timed or semi-actuated signal control environments. The traffic signal states are tractable for the CV to make decisions in these cases. Besides, some researches focus on situations with unknown traffic signal timing. For example, Mahler and Vahidi [25] developed a simple phase prediction module that uses historically-averaged data and real-time phase data. Then, the ego vehicle is controlled with probabilistic traffic signal circumstances by DP. Sun et al. [26], who formulated a random variable called effective red-light duration and appended a chance constraint to the optimal control problem. The control law is also determined by DP in their work. Nevertheless, the above-mentioned methods assume that the traffic light operates cyclically with a fixed cycle and duration. The situation with adaptive traffic signal control (ATSC) is ignored by scholars.

Meanwhile, the traditional traffic signal control methods will be gradually eliminated in some urban areas with the development and construction of smart cities. ATSC is becoming a promising way to alleviate mounting traffic congestion. As an important part of an intelligent traffic system (ITS), ATSC approaches change the signal state in real-time according to the traffic flow with the intent to reduce vehicle delay [27]. In particular, the reinforcement learning (RL) technique is one of the most up-and-coming approaches to implementing ATSC in terms of its learning-based mechanism. By combining with the deep learning method, deep reinforcement learning (DRL) shows great potential to

solve decision-making problems like ATSC [28,29]. As far as the application is concerned, the DRL-based traffic signal control method is accomplished in the real world by fusing multi-source data. The application of the ATSC in Shenzhen, China, brings about more smooth traffic and less vehicle delay [30].

ATSC methods represented by DRL algorithms have two characteristics: (1) the change of the signal phase is determined by the current traffic state, where the state can be the average waiting time of vehicles, queue length, or some other numerical features; (2) the traffic lights controlled by ATSC methods do not have fixed cycles and phase plans. In this case, the standard SPaT information cannot be produced and obtained by CV, and this leads to more uncertainty regarding the control of the vehicle. More specifically, if the intelligent driving system of the vehicle cannot know when the phase would change, it is difficult to control the vehicle to cross the intersection during the green light, and this may raise the possibility that the vehicle performs a stop-and-go motion with increased energy consumption. Concurrently, considering the surrounding cars and the dynamics of the ego vehicle itself, the eco-driving task in an ATSC environment becomes a complex nonlinear optimal control problem, which is hard to be solved numerically. Therefore, the previous studies can no longer be applied to such situations. However, considering that the ATSC will replace the position of a traditional fixed timing signal control, the eco-driving strategy in such a situation is needed.

While model-based control strategies like DP and MPC are time-consuming, the model-free DRL algorithms are capable of controlling the vehicle in a real-time way with a “trial-and-error” mechanism. In DRL theory, the agent can learn a policy and carry out the action produced by the policy according to the observed state. Thus, the DRL algorithms are applied to the eco-driving domain to realize adaptive control of vehicles. In this case, Shi et al. [31] utilized Deep Q Network (DQN) to build an eco-driving system, but the action of the agent of the set is a discrete acceleration rate, since DQN cannot deal with continuous action space. Similarly, the framework proposed by Mousa et al. [32] encountered the same problem when the simplified action space leads to a local optimum solution. To address this issue, Guo et al. [33] introduced a deep deterministic policy gradient (DDPG) approach to control the ego agent with continuous acceleration. Furthermore, a twin delayed deep deterministic policy gradient (TD3) algorithm-based automated eco-driving method is reported by Wegener et al. [34], while TD3 is an improved version of DDPG. Although the adaptive control can be implemented in these DRL-based approaches, the accurate SPaT information is taken as part of the state of the agent. As these approaches have tested the effectiveness of the DRL-based control in a fixed timing situation [35,36], they cannot be directly applied in the ATSC environment. Furthermore, to the best of the authors’ knowledge, all of the previous DRL-based eco-driving system works lack explicability, which means that we cannot know why the ego vehicle accelerates or decelerates in every specific time step. It is obvious that the black-box logic may give rise to some safety-related issues and have a negative impact on reliable control.

In this paper, we focus on electric vehicles, which attract the public’s attention due to its eco-friendly feature, benefitting from the advancements of battery-related research [37,38]. We also develop a Markov Decision Process (MDP) model for the eco-driving problem in the ATSC traffic environment to fill the previous technical gap. The optimal speed profile is generated by the Proximal Policy Optimization (PPO) algorithm, which is capable of solving continuous action space problems. Meanwhile, SHapley Additive exPlanations (SHAP) [39] is used to explain the decision process of the agent (i.e., the ego vehicle) to make the DRL algorithm-based approach explainable.

The main contributions of our paper are summarized as follows:

- (1) A general MDP for eco-driving control is established, whereas the reward function of the system considers energy consumption, traffic mobility, and driving comfort;
- (2) Unlike operating in a fixed-timing traffic light environment, the system can be dedicated to a DRL-based ATSC environment, while the proposed state sharing mechanism allows the ego vehicle to take the state of the traffic light agent as part of the state itself;

- (3) Whereas conventional optimization-based methods suffer from computation complexity, a model-free DRL algorithm is adopted to generate the speed profile of the vehicle in real-time, and the SHAP approach is used to explain and visualize the decision of the ego vehicle, which is of help to reveal the black-box decision process of the DRL agent.

The remainder of this article is structured as follows. Section 2 introduces the preliminaries of DRL and DRL-based ATSC. Section “Problem formulation” provides the formulation of the eco-driving control problem with ATSC. Section 4 proposes the MDP framework and PPO algorithm with the intent to solve the problem. Section 5 reports a series of simulations carried out in the SUMO platform [40] to demonstrate the feasibility and effectiveness of the proposed method. Finally, some concluding remarks are presented in Section 6.

2. Preliminary

2.1. Deep Reinforcement Learning

Reinforcement learning is motivated by the MDP framework, which can be defined as a five-tuple (S, A, R, P, γ) . The process can be depicted by the agent-environment interface, which is shown in Figure 1. In a fully-observed MDP, the agent can observe the current state of the environment $s_t \in S$, while the environment can be a simulator or mathematical model. The agent will conduct action $a_t \in A$ and receive the reward $r_t \in R$. The actions of the agent directly influence its subsequent cumulative rewards G_t for a K -step control problem, which can be computed according to $G_t = r_{t+1} + \gamma r_{t+2} + \dots + \gamma^{\tau-1} r_{t+\tau} = \sum_{\tau=0}^K \gamma^\tau r_{t+\tau+1}$. The parameter γ is a discount factor to measure the weight of the current reward and future reward. Moreover, P is a function specifying the transition probability for the agent conducting the action a_t in state s_t .

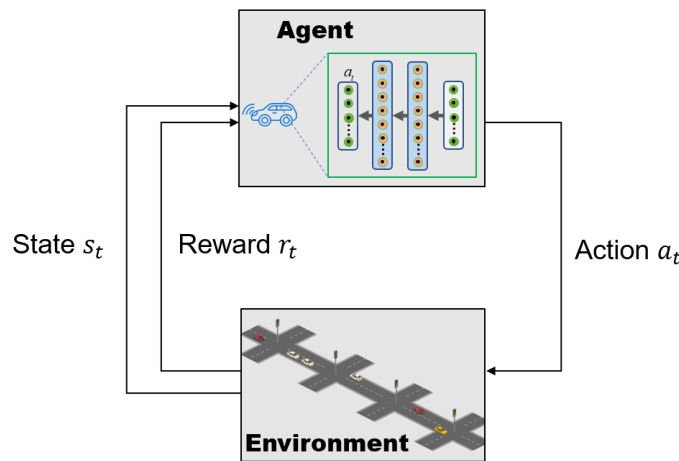


Figure 1. Agent-environment interface for fully-observed MDP.

The goal of the agent is to maximize the expected cumulative reward in each timestep t . Thus, an optimum policy $\pi(a|s) : S \times A \rightarrow [0, 1]$ is supposed to be learned by the agent to guide the agent to select the optimal action a_t with state s_t to maximize its return. Following the policy π , the state-action (i.e., Q-function) can be given by:

$$Q^\pi(s_t, a_t) = \mathbb{E}_\pi \left(\sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}(s_{t+\tau}, a_{t+\tau}) | s_t, a_t \right). \tag{1}$$

In this case, an optimal policy for the agent is:

$$\pi_*(s) = \underset{a \in A}{\operatorname{argmax}} Q_*(s, a), \quad s \in S, \tag{2}$$

which can generate the optimal action for the agent in each step.

However, the space of the state and action can be large, and the high-dimension data will lead to the “curse of dimensionality” in the process of learning. Deep neural networks are introduced in the RL domain to make an approximation as a result. We assume that there is a neural network with parameters θ , and an approximate optimum policy $\pi_*(s, \theta)$ is established. Finally, the agent needs to learn in a data-driven way to update θ in order to make the equation $\pi_*(s, \theta) \approx \pi_*(s)$ come true.

2.2. Adaptive Traffic Signal Control

ATSC is a widely studied concept in the past few years. It is usually accomplished by fuzzy logic, swarm intelligence, or neural network [41]. As the traffic lights are supposed to respond to the changing traffic flow, the prevailing DRL framework brings about better solutions to the problem. In this paper, we implement a DRL-based ATSC system to demonstrate the feasibility of our framework, but it should be noted that the methodology is true for most of the state-dependent ATSC approaches.

The RL-based ATSC also follows the MDP model, and the corresponding definition is as follows:

- (1) *State*: The first part of the state for a traffic light is represented by the queue length ratio in each lane. We assume that the length of homogeneous vehicles is denoted as l_v , and the minimal gap between vehicles is ω . Then the time-dependent queue length ratio of traffic light j can be defined as:

$$\Psi_j(t) = \min\left(1, \frac{n_i(t)(l_v + \omega)}{L_i}\right), i \in I_j, \tag{3}$$

where, I_j is the set of the lanes connected to the intersection area, while n_i, L_i is the number of halting vehicles (i.e., the cars with a speed less than 0.1 m/s^2) and the length of the lane, respectively.

The second part of the state for a traffic light is the one-hot encoding of its current phase with a predefined phase scheme. Let the encode vector at time t be $E_j(t)$, and the state of traffic light j is $s_j(t) = [\Psi_j(t), E_j(t)]$.

- (2) *Action*: The action in this paper is defined as $a = 1$: switch to the next phase, and $a = 0$: maintain the current phase. The action is carried out with the interval of 25 s, while the specific definition of the signal phases is given in Figure 2. It should be noted that a 3 s yellow phase will be inserted if the phase changes.
- (3) *Reward*: The reward of the traffic signal agents is defined as the summation of the waiting time of vehicles on the lanes set I_j . The definition aims to reduce the average waiting time of vehicles to improve traffic efficiency.

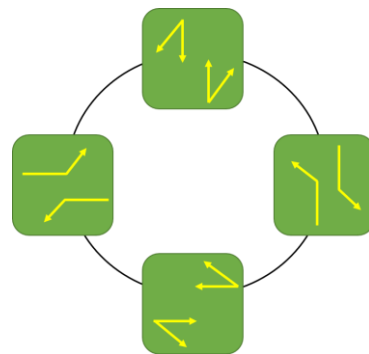


Figure 2. The phase diagram for the traffic signals in this paper.

We adopt the RAINBOW algorithm to control the traffic light, whereas the algorithm uses several tricks to improve the performance of the original DQN. The details of the algorithm can be found in [42]. In addition, a signalized corridor is taken as the research

object, which means that the multi-agent system is built for the traffic lights. The training process is shown in Figure 3. The figure manifests the change in the total reward of the multi-traffic signal agents, and a distinct convergence trend is shown, which demonstrates the effectiveness of the training algorithm.

More details of the implementation of the DRL-based ATSC can be found in Appendix A, as it is not the core part of the presented paper.

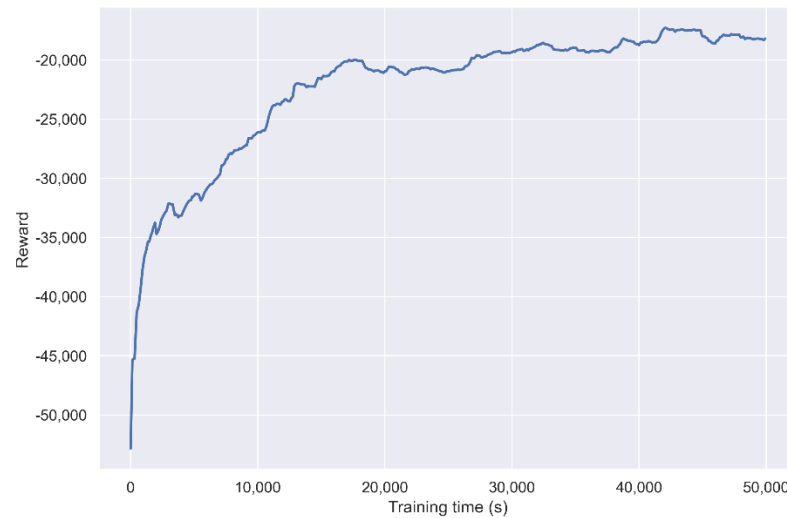


Figure 3. The training process of the traffic lights with RAINBOW.

3. Problem Formulation

The eco-driving problem can be formulated as an optimal control problem, whether in a fixed timing signal environment or an ATSC environment. The general scenario for the ego vehicle crossing a single intersection is shown in Figure 4.

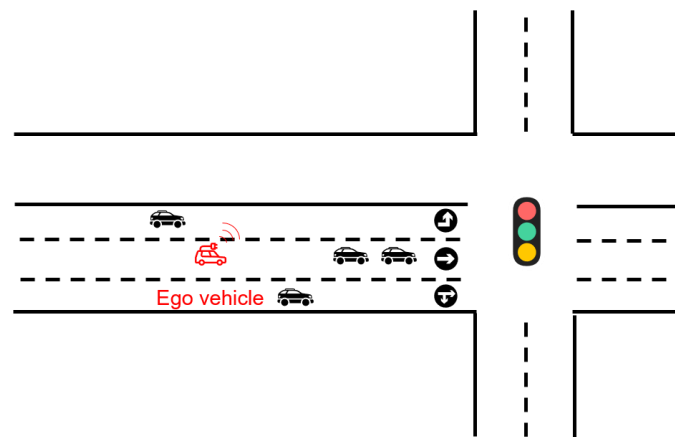


Figure 4. The driving scenario for the ego vehicle in a single intersection.

Some assumptions are made in this paper for simplification without losing generality:

- (1) The ego vehicle knows its position on the lane, speed, and acceleration;
- (2) The data packets can be transmitted by the traffic light to the ego vehicle through V2I communication without delay or packet loss;
- (3) The acceleration/deceleration of the ego vehicle can be controlled accurately with the onboard control module;
- (4) The ego vehicle will not change the lane, which means that only longitudinal movement is considered.

With the intent to describe the problem, we also formulate the optimal control problem form for the task, while the control item is the acceleration of the ego vehicle.

Firstly, the main goal of the problem is to reduce the energy consumption in the whole process:

$$J_1 = \sum_{j=1}^m \int_{t_0^j}^{t_f^j} \phi(v(t), a(t)) dt, \tag{4}$$

where m is the total number of intersections; t_0^j is the beginning time for the ego vehicle to enter the lane of intersection j , while t_f^j refers to the time that the ego vehicle arrives at the stop line of the junction; ϕ is the energy function that can produce instantaneous energy consumption of the vehicle. In this paper, the energy model for electric vehicles is embedded in SUMO with an energy brake-recovery mechanism, and the details can be found in [43]. Moreover, $v(t)$ and $a(t)$ represent the speed and acceleration of the vehicle in time t .

Secondly, mobility should also be considered, and we measure that by total travel time:

$$J_2 = \sum_{j=1}^m \frac{v_{max}(t_f^j - t_0^j)}{L_j}, \tag{5}$$

where v_{max} denotes the maximal road speed limit.

Finally, a soft constraint is imposed on the objective function, and the objective function is denoted as:

$$J(a) = \omega_1 J_1 + \omega_2 J_2 + \sum_{j=1}^m c(j, t_f^j) \frac{1}{\epsilon + p(j, t_f^j)}, \tag{6}$$

where ω_1 and ω_2 are the weighting parameters; $c(j, t_f^j)$ is a boolean value: the value is set to 0 when the traffic light is green; the value is set to 1 if the vehicle is not allowed to cross the junction owing to the spatiotemporal interval produced by a traffic signal. ϵ is a small number to serve as a penalty item and $p(j, t_f^j)$ measures the green probability for traffic light j in time t_f^j . This soft constraint encourages the vehicle to pass the road crossing without halting in front of the stop line because of the red light, and the energy can be saved in this case.

Therefore, let $d_j(t)$ be the lane position of the vehicle approaching traffic light j , and $x(t) = [d_j(t), v(t)]^T$. The whole control problem is formulated as:

$$\min J(a). \tag{7}$$

Subject to:

$$\dot{x}(t) = [v(t), a(t)]^T, t \in [t_0^1, t_f^m], \tag{8}$$

$$0 \leq v(t) \leq v_{max}, t \in [t_0^1, t_f^m], \tag{9}$$

$$d_{max} \leq a(t) \leq a_{max}, t \in [t_0^1, t_f^m], \tag{10}$$

$$a(t) \leq a^{CF}(t), t \in [t_0^1, t_f^m], \tag{11}$$

$$x(t_0^1) = x_0, x(t_f^m) = x_f, \tag{12}$$

$$d_j(t_f^{j-1}) + D_j = d(t_0^j), i \in [2, m]. \tag{13}$$

Equations (9) and (10) denote the dynamics constraints of the ego vehicle. (11) denotes the safety consideration, while $a^{CF}(t)$ is the value of acceleration generated by the car-

following model. The acceleration of the ego vehicle generated by the control model cannot go beyond the car-following model-based value to ensure that the ego vehicle will not collide with the potential leader cars. (12) denotes the start state and the end state. In fact, the end speed can be any value. Finally, (13) denotes the distance connection between multi road segments, while D_j is the cover distance of the ego vehicle travel in inside junction j .

Obviously, we need to establish a function to fit $p(j, t_f^j)$ in real-time during the process of control. Then, some methods like DP can be applied to get the policy. However, it is difficult to finish that because the accurate prediction cannot be easily achieved in an ATSC environment. Meanwhile, the control requires high real-time performance in which the model-based DP cannot satisfy. Thus, we try to accomplish the adaptive optimal control through the DRL technique instead of some brute force ways.

4. Methodology

As shown in Figure 5, the sensed traffic state is taken as the input of traffic signal agents in an ATSC environment. Since the accurate change rule of the traffic signal phase is not clear, the vehicle agent is supposed to understand the dynamics of the traffic signal agents. From this point of view, the input state of the traffic signals can be part of the state of vehicle agents, utilizing the V2I communication to transmit the information. In addition, some other related states are collected to form the decision basis of the ego vehicle. The comprehensive design of the state vector of the CV, which is enhanced by the state sharing mechanism between the heterogeneous agents, is promising to help the CV catch the regular pattern of the adaptive traffic signals.

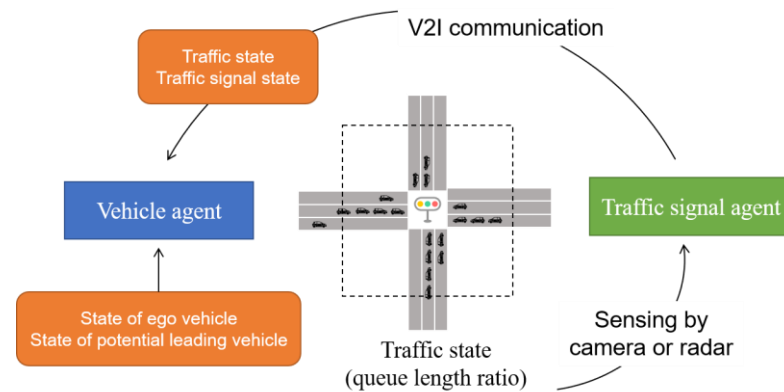


Figure 5. The state description of each kind of agent.

4.1. MDP Model

Similar to the RL-based ATSC, the MDP model is a prerequisite for applying the RL algorithm. Based on the assumptions we made in the previous section, we can specify the MDP according to the optimal control problem we build to make a mapping. The state, action, and reward of the MDP is defined as follows:

- (1) *State*: The ego vehicle selects action in terms of the state, so the definition of the state is crucial. The state of this problem should include its dynamics, the state of its surrounding vehicles, and the state of the traffic lights. Therefore, a vector with multi-elements is built to serve as the state of the ego vehicle:

$$s^v(t) = [d^s(t), v(t), \Delta d(t), \Delta v(t), \Delta a(t), \xi_j, t_j^c, s_j(t)]^T, \tag{14}$$

where $d^s(t)$ measures the distance from the vehicle to the stop line, whereas the agent needs to know how far to enter the intersection; $\Delta d(t)$, $\Delta v(t)$, and $\Delta a(t)$ denotes the distance difference, speed difference, and acceleration difference between the potential leading vehicle and the ego vehicle. This may help the ego vehicle keep a safe distance between itself and the front vehicle or prevent itself from entering the tail of a queue. More precisely,

the front vehicle can be the tail of the queue, so the ego vehicle may enter the queue and keep the stop-go movement, which may increase the energy consumption significantly. However, a leading vehicle does not always exist, so we set the judgment threshold $\rho = 300$ m to measure the relative distance between the front vehicle and the ego one. Let the absolute position of the front vehicle be $d^f(t)$, and the value of the ego vehicle be $d^e(t)$, then we have:

$$\Delta d = \begin{cases} d^f(t) - d^e(t), & \text{if } d^f(t) - d^e(t) < \rho \\ \rho, & \text{otherwise} \end{cases}, \tag{15}$$

when $d^f(t) - d^e(t) < \rho$, the value of $\Delta v(t)$ and $\Delta a(t)$ can be calculated in the same way. Otherwise, the two values will be set to the default values Δv_d and Δa_d .

Moreover, ζ_j in (9) is a boolean value that represents whether the lane where the vehicle is located is opened with the green light in the current phase; t_j^c denotes the time to act for the first downstream traffic signal agent. Finally, $s_j(t)$ is the state of traffic light j (i.e., the traffic light can share its state with the ego vehicle through V2I communication). The three groups of the parameters here consist of the state related to the traffic signal agent, which may help the vehicle agent understand the performance of the traffic lights.

This state definition applies regardless of whether the vehicle is on a road segment or inside an intersection. Hence, the framework can be used for a multi-intersection scenario naturally.

- (2) *Action:* The action is consistent with the control item in the optimal control problem, that is, the acceleration of the ego vehicle. The action space should satisfy Equation (10) to generate an effective value. Meanwhile, the speed of the agent should also satisfy (9), which means that the speed change of the vehicle is denoted as:

$$v(t) = \max(\min(v_{max}, v(t-1) + \hat{a}_t), 0), \tag{16}$$

where a_t is the output of the DRL algorithm. However, the value a_t cannot be taken as the final acceleration to impose on the vehicle because (11) should also be accomplished. Accordingly, the amended acceleration is:

$$a_t = \min(\hat{a}_t, a^{CF}(t)), \tag{17}$$

where the $a^{CF}(t)$ is determined by IDM [44] in this paper, and the parameters of the car-following model are set to default values in SUMO to enhance traffic safety.

- (3) *Reward:* The definition of the reward function should be similar to the cost function (6) to get the same control effect. However, the travel time item in (6) is a Meyer item, which can only be calculated at the end of the process, in other words, the value can be obtained only if the ego vehicle reaches the stop line. In our RL framework, it is difficult to allocate the credit to every step because the contribution to the travel time of each step is hard to be measured. Thus, we take the travel distance in each step to replace that value, which is denoted as $l^d(t)$. Besides, we add jerk, which is defined as the differential of the acceleration, to the reward function with the intent to improve the driving comfort. The reward function is denoted as:

$$r^v(t) = -\omega_1\phi(v(t), a(t)) + \omega_2l^d(t) - \omega_3\dot{a}(t) - \delta\frac{1}{\epsilon}, \tag{18}$$

where δ is a boolean flag to identify whether the ego vehicle is halting. In this case, a penalty value $\delta\frac{1}{\epsilon}$ will be added if the vehicle encounters a red light and wait. Meanwhile, this item can encourage the agent not to join a queue. It should be noted that $\omega_1\phi(v(t), a(t))$ can be a negative value owing to the recycling mechanism of the braking energy of the electric vehicles. The reward function considers energy saving, traffic mobility, and comfort simultaneously, which is practical and economical.

4.2. Proximal Policy Optimization

PPO is an on-policy reinforcement learning algorithm, which can be used for problems with continuous action spaces [45]. Generally, the on-policy interaction takes the return in each step to estimate the long-term return $\sum_{\tau=0}^K \gamma^\tau r_{t+\tau+1}$, and this will lead to large variance. Accordingly, some studies use an actor-critic framework with an advantage function to estimate the return, but it will increase the bias. Therefore, the paper applies generalized advantage estimation (GAE) to optimize the advantage function to get a trade-off between variance and bias.

The movement of the ego vehicle can be optimized when the parametric policy $\pi(\theta^p)$ approaches the optimal policy. In this case, the main goal is to find the optimal parameter set θ_*^p of the neural network. That means the objective function should be maximized:

$$R(\theta^p) = \mathbb{E}_{\eta \sim \mathcal{D}_k} \left[\sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}(s_{t+\tau}, a_{t+\tau}) \right], \tag{19}$$

in which the $\mathcal{D}_k = \{\eta_i\}$ is a set of transitions collected from the interaction between the agent and the environment. The transition sequence η_i herein refers to a set consisting of $\{s_1, a_1, r_2, s_2, a_2, r_3, \dots\}$, and a collection of transitions can be used in training.

The training process is shown in Algorithm 1. PPO makes use of a value function approximation (VFA) V_{θ^v} , which also takes an artificial neural network (ANN) with a parameter set to θ^v to estimate the expected total reward in the future. Over each epoch $k = 1, 2, 3, \dots, K$, the parameters of the neural networks can be updated. The parameter θ is learned in the following way:

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} E_{s, a \sim \pi_{\theta_k}} [L(s, a, \theta_i, \theta)], \tag{20}$$

where $L(s, a, \theta_k, \theta)$ is given by:

$$L(s, a, \theta_k, \theta) = \min(\beta_k(\theta) A^{\pi_{\theta_k}}(s, a), g(\varepsilon, A^{\pi_{\theta_k}}(s_t, a_t))), \tag{21}$$

where the definition of $\beta_k(\theta)$ and $g(\varepsilon, A^{\pi_{\theta_k}}(s_t, a_t))$ is given by Equations (22) and (23):

$$\beta_k(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, \tag{22}$$

$$g = \operatorname{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, 1 - \varepsilon, 1 + \varepsilon\right) A^{\pi_{\theta_k}}(s, a). \tag{23}$$

The parameter ε is preset to stipulate an interval $[1 - \varepsilon, 1 + \varepsilon]$ that defines the upper and lower bound of $\beta_k(\theta)$. The purpose of (23) is to restrict the search of θ_{k+1} within a predefined range that is close to θ_k . The notation $A^{\pi_{\theta_k}} \in \mathbb{R}$ is the value produced by an advantage function reflecting the value of a selected action in accordance with current policy π_{θ_k} concerning other feasible decisions [46]. Moreover, the clip operation output $(1 + \varepsilon)A^{\pi_{\theta_k}}$ if $A^{\pi_{\theta_k}} \geq 0$, otherwise $(1 - \varepsilon)A^{\pi_{\theta_k}}$ will be exported. In this paper, GAE is taken to calculate $A^{\pi_{\theta_k}}$:

$$A^{\pi_{\theta_k}} = \lambda_k + (\gamma\chi)\lambda_{k+1} + \dots + (\gamma\chi)\lambda_{K-1}, \tag{24}$$

where the discount factor $\gamma \in [0, 1]$ and the GAE parameter $\chi \in [0, 1]$ are used to balance the estimated biases and variances. The notation λ_k is the one-step temporal difference (TD), which is defined as:

$$\lambda_k = \begin{cases} r_{k+1} - V_{\theta^v}(s_t), & \text{if } k = K - 1 \\ r_{k+1} + \gamma V_{\theta^v}(s_{t+1}) - V_{\theta^v}(s_t), & \text{otherwise} \end{cases}. \tag{25}$$

During the training process, both θ^p and θ^v can be updated iteratively through the Adam optimization algorithm [47].

Algorithm 1 PPO for eco-driving training

```

1 Initialize policy parameters  $\theta^p$  and value function parameters  $\theta^v$  with random values
2 for  $k = 1, 2, 3 \dots k$  do
3   Carry out the preloading process and insert the ego vehicle into the road network
4   Control the vehicle according to policy  $\pi(\theta^p)$  and collect the set of trajectories  $\mathcal{D}_k = \{\eta_i\}$ 
5   Compute reward-to-go value  $G_t$ 
6   Compute advantage estimates  $\hat{A}_t$  (optimized by GAE) based on the current value function  $V_{\theta^v}$ .
7   Update the policy by maximizing the objective via Adam optimizer:
 $\theta_{k+1}^p = \operatorname{argmax}_{\theta^p} \frac{1}{|\mathcal{D}_k|T} \sum_{\mathcal{D}_k} \sum_{t=0}^T \min \left( \frac{\pi_{\theta^p}(a_t|s_t)}{\pi_{\theta_k^p}(a_t|s_t)} A^{\pi_{\theta_k^p}}(s_t, a_t), g(\varepsilon, A^{\pi_{\theta_k^p}}(s_t, a_t)) \right)$ 
8   Fit value function by regression to minimize mean-squared error via Adam optimizer:
 $\theta_{k+1}^v = \operatorname{argmin}_{\theta^v} \frac{1}{|\mathcal{D}_k|T} \sum_{\mathcal{D}_k} \sum_{t=0}^T (V_{\theta^v}(s_t) - R_t)^2$ 
9 end for

```

4.3. SHapley Additive exPlanations

We might want to know why the vehicle accelerates or decelerates under the driving of the PPO algorithm. In other words, the agent observes the state that consists of some elements. Some of the elements may spur the vehicle to accelerate, while others may slow down the vehicle. Meanwhile, the attribution of the features (i.e., the element in the state vector) is different for the agent to make decisions. Considering safety and reliability, we should build an explanation module to describe the behavior of the agent.

It is known that feature attribution is a common approach to analyzing a trained machine learning model, whereas the RL is a branch of machine learning. Conventionally, the machine learning community takes tree-based models as one kind of the interpretable methods [48]. Nevertheless, this paper deploys a DRL model, which can only be explained in a post-hoc manner. We assume that F is a machine learning model that maps some input $x = (x_1, \dots, x_n) \in R^n$ to the target values. The feature attribution of the prediction at x relative to a baseline input x_b can be denoted with a vector $A_F(x, x_b) = (fa_1, fa_2, \dots, fa_n) \in R^n$ where fa_i represents the contribution of x_i to the prediction $F(x_i)$. In this case, the Shapley value becomes an effective method to distribute the total gains of a cooperative game to a coalition of cooperating players [49]. More specifically, taking a coalitional game with n players and a function F that maps the subset of players to the real numbers $f : 2^N \rightarrow R$ as an example, the attribution amount of player i to the game is:

$$A_f(i) = \sum_{C \subseteq N \setminus \{i\}} \frac{|C|!(n - |C| - 1)!}{n!} (f(C \cup \{i\}) - f(C)), \tag{26}$$

where $f(C)$ is the worth of coalition C , illustrating the total expected sum of payoffs the members of C can derive from the cooperation. The gist of the equation is: the difference between value function with and without player i is computed for each coalition C without player i , so the contribution of player i is equal to the weighted difference $\frac{|C|!(n - |C| - 1)!}{n!}$. For the RL-based eco-driving problem, we formulate a game for the control state at each timestep. The “total gains” denote the acceleration of the vehicle, while the “players” is the set of the state in the MDP model. The Shapely-value-based approach tends to approximate Shapley values by examining the effect of removing an element under all possible combinations of the presence or absence of the other elements in the state vector.

However, computing the Shapely value for every feature is still challenging. On the other hand, LIME (Local Interpretable Model-Agnostic Explanations) [50] is proposed to be an additive feature attribution method that learns an explainable model like a decision tree or linear regression model $w(z')$ to approximate the target black-box model locally, where z' is a simplified binary input vector. Then the simplified features are transformed into the original algebraic space so that the corresponding target value is computed by $w(z') = f(h_x(z'))$, where h_x is a function that maps the simplified input to the original input. The Kernel SHAP method draws lessons from LIME and computes the weight for

each element in z' . A SHAP value estimates the change in predicted output, from the base value $E[f(x)]$ to $f(x)$, attribute to every single feature i . More precisely, the SHAP value can be $E[f(z')|z'_1 = x_1] - E[f(z')]$ for feature with $i = 1$.

5. Simulation Analysis

5.1. Simulation Configuration

The simulations are carried out on the SUMO platform. As an open-source software, SUMO provides higher privileges to do further development based on its framework. Meanwhile, SUMO can interact with Python programs through the Traci interface, whereas our algorithm and training procedure are coded in Python.

A signalized corridor scenario is established in SUMO, which can be shown in Figure 6. The lane configuration for each intersection is the same as Figure 4. For each episode of train or evaluation, the traffic will be loaded in the first 100 s with a random arrival rate. Then the ego vehicle is inserted into the leftmost lane. The preloading procedure is to simulate the real traffic situation, that is, the ego vehicle can be influenced by some other vehicles. After that, the ego vehicle will cross the subsequent five signalized intersections with the support of the PPO algorithm. An episode of simulation will be terminated when the ego vehicle has crossed the last junction. Moreover, the transitions of each episode of simulation will be recorded and saved to prepare for the explanation process based on SHAP.

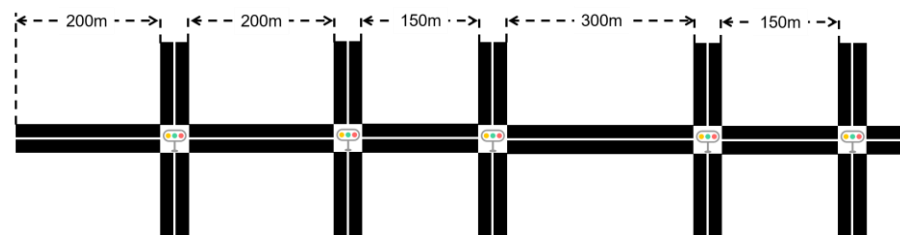


Figure 6. The simulation scenario.

Table 1 presents the general parameter settings for simulations. The parameters related to PPO are tuned manually through a series of simulations. We use a fully connected neural network architecture for both parametric policy and parametric value function. The units of the neural networks with three hidden layers are [512, 256, 64]. In addition, the default settings of weighting parameters ω_1 , ω_2 , and ω_3 are set to 5, 3, and 7. The sensitivity analysis of these parameters is presented after the general analysis.

Table 1. General parameter settings for simulations.

Symbol	Description	Value
v_{max}	Road speed limit	13.89 m/s
a_{max}	Maximal acceleration of vehicles	3.0 m/s ²
d_{max}	Maximal deceleration of vehicles	−4.5 m/s ²
Δv_d	Default value for speed difference	−14 m/s
Δa_d	Default value for acceleration difference	−7.2 m/s ²
ϵ	The value corresponds to the penalty item in Equation (18)	0.01
χ	The GAE parameter	1.0
—	Learning rate of the Adam optimizer	0.0001
ϵ_1	Clip parameter for policy	0.3
ϵ_2	Clip parameter for value function	10.0

5.2. Simulation and Discussion

The simulations are built upon the environment described in the last subsection. We introduce the RAINBOW algorithm to be the baseline and make comparisons with PPO.

The action space of the RAINBOW agent is set to a 16-dimension vector because the value-based algorithm can only be applied to problems with continuous action space. We divide the acceleration of the ego vehicle between -4.5 m/s^2 and 3.0 m/s^2 in steps of 0.5 m/s^2 ; the agent selects an action from the 16-dimension action space, and we will see that the optimal control cannot be implemented in this discrete-action-space way.

The scatters in Figure 7 show the data point of different metrics during the training processes of PPO and RAINBOW. In addition, the line plots displayed above the scatters are the linear regression result of the corresponding data. It can be found that the energy consumption of the RAINBOW agent cannot be reduced significantly, despite the reward increases. Compared with RAINBOW, the energy consumption of the PPO agent converges to a nominal value, with extra sacrifice in terms of the travel time. The training profiles illuminate that the proposed PPO-based framework is capable of searching for a solution that enables the vehicle to run in a more energy-saving way.

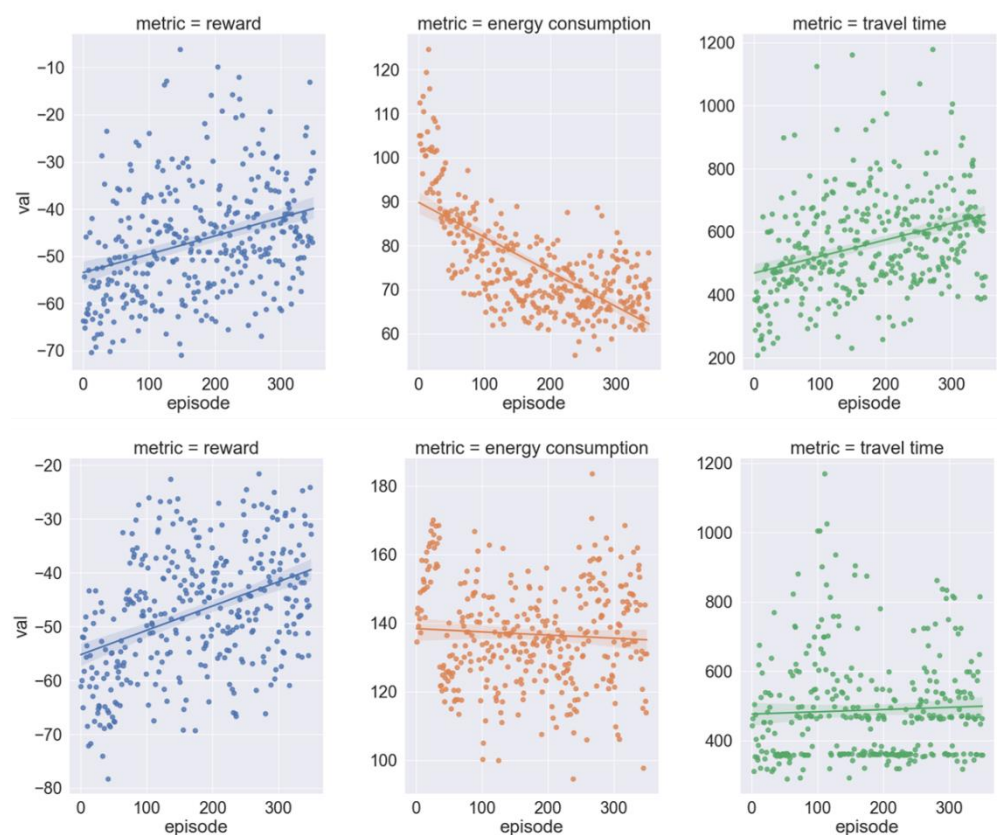


Figure 7. The change of reward, energy consumption, and travel time during the training processes of PPO and RAINBOW.

After the training procedure, the well-trained agents are utilized to evaluate their performance. The results are collected from 20 independent simulations, and the random seed is set to the same value for the same round simulation of different control methods. Figure 8 illustrates the evaluation results of PPO, RAINBOW, and IDM, whereas the IDM can reflect the general performance of human drivers. According to Figure 8a, we can see that the RL-based approaches will reduce the mobility of the ego vehicle. However, PPO can save electric energy to a great extent compared with IDM, which can be seen in Figure 8b. On the other hand, although the RAINBOW sacrifices traffic mobility, the energy consumption still remains at a high level. The result proves that the DQN-based algorithm cannot deal with the eco-driving problem in an ATSC environment. Moreover, the mean jerk is improved slightly under the control of PPO compared with the situation of IDM, and this can be caused by some braking behaviors with the intent to recover the energy. Thus, the overall comfort of driving under RL-based control may decrease slightly. On

average, PPO can save 41.06% energy consumption compared with manual driving (i.e., IDM) in an ATSC environment.

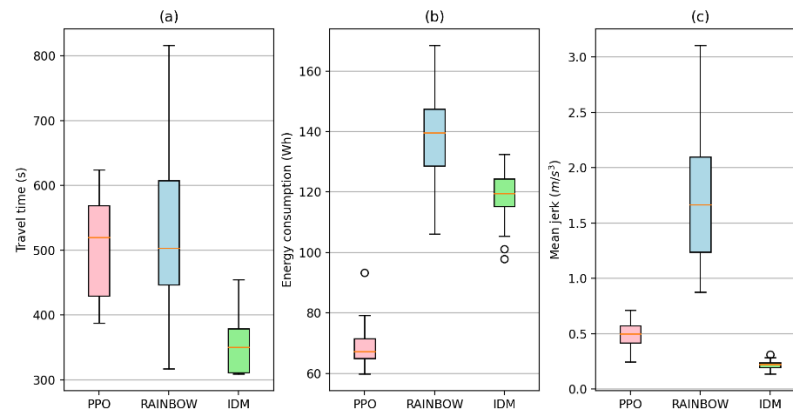


Figure 8. The evaluation results for PPO, RAINBOW, and IDM. (a) Travel time; (b) Energy consumption; (c) Mean jerk.

Figure 9 presents the results in terms of electricity consumption for 20 groups of independent simulation-based evaluations in detail. The figure demonstrates that the proposed method lowers the consumed energy steadily compared with IDM, which is deemed as representative of normal human driving behavior. By contrast, the results produced by RAINBOW can be even worse than the baseline in most cases, further proving that the value-based algorithm with discrete action space cannot fit into the environment with high uncertainty.

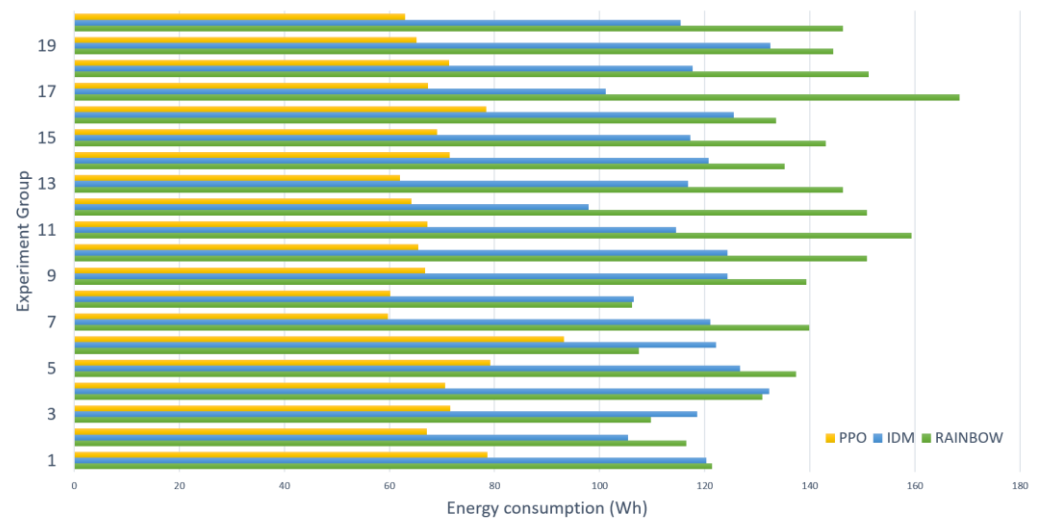


Figure 9. Energy consumption comparison for 20 groups of experiments.

We elicit the explain module in the methodology section because we are interested in the decision process of the agent. Here, we can analyze the feature attribution of the element in the state vector. Two samples in the evaluation results are selected randomly; the SHAP value is calculated for each simulation step. The trajectories of the ego vehicle are plotted in Figures 10 and 11, and then we mark several trajectory parts with black square boxes. The SHAP force plot is provided for the boxes by using the tool developed by Ref. [51]. The features in Figures 10 and 11 corresponds to the state definition $s^v(t)$: “dist2stop” corresponds to $d^s(t)$; “is_green” corresponds to ζ ; “time_to_change_phase” represents t^c ; the features started with “obs_ts” are part of the state of the traffic signal

agent $s_j(t)$. The red part of the SHAP charts means that the corresponding features drive the vehicle to speed up, while the blue part drives the vehicle to slow down.

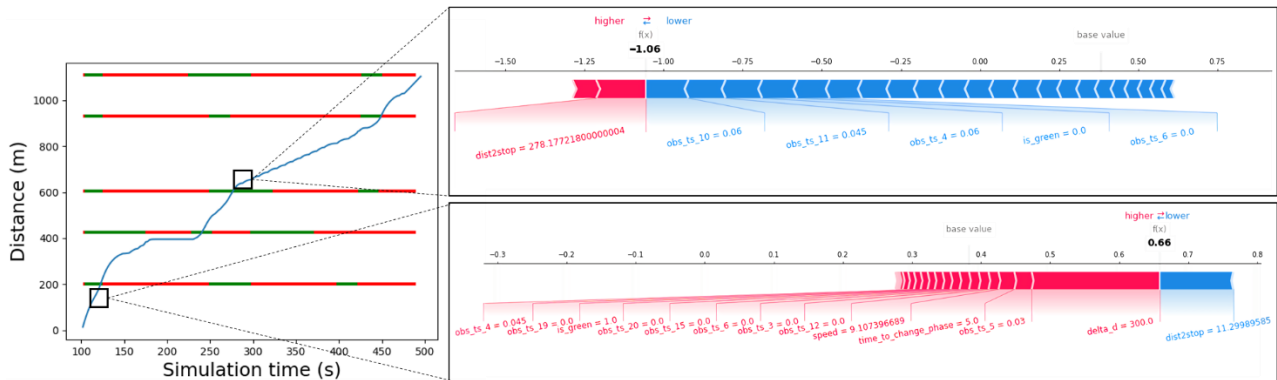


Figure 10. Sample trajectory 1 and its SHAP representation.

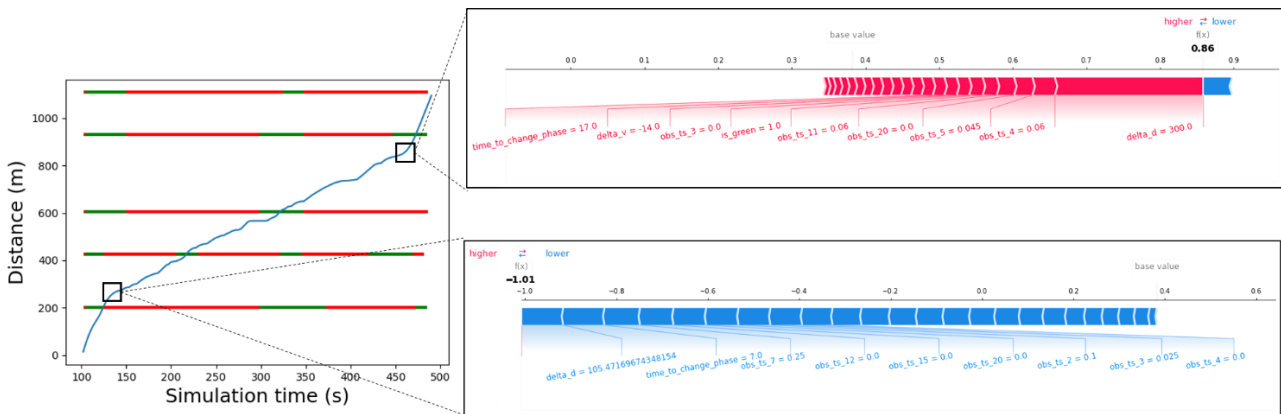


Figure 11. Sample trajectory 2 and its SHAP representation.

We can analyze the trajectories depicted in Figures 10 and 11. It can be found that the agent may stop at the proximity of the intersection due to the uncertainty caused by the queue and the traffic light. This implies that the optimal control law cannot be given by the proposed algorithm. However, electrical energy can be saved significantly even in such an uncertain situation. The agent can make effective decisions through the use of the obtained information. For example, the upper box in Figure 10 manifests a slow-down decision of the ego vehicle. In this case, the feature $d^S(t)$ encourages the agent to accelerate, but the state of the traffic light lets the agent know that the traffic light will not turn green soon. Hence, the agent chooses to slow down to avoid the red light. The same process is true of the lower box in Figure 10; the traffic signal has 5 s to switch, and the ego vehicle judges that the green light will not continue according to the traffic light state. Therefore, the agent chooses to accelerate on the premise that there is no car ahead.

Taking Figure 11 as an example, more quantitative analysis can be conducted. The “base value” in the SHAP chart denotes the mean value of the acceleration in the whole episode of the simulation. The acceleration of the upper box in Figure 11 is 0.86 m/s^2 , which is the result of the incentive effect of the features. Similarly, almost all of the features prevent the agent from accelerating for the situation in the lower box, so the action of the agent is -1.01 m/s^2 . Meanwhile, the base values in the sampled trajectories are small. It means that although the trajectory of the vehicle is not so smooth, the acceleration and deceleration range of the vehicle is relatively small. This can help the vehicle save energy and improve comfort.

5.3. Sensitivity Analysis

The sensitivity analysis is carried out by exploring the effect of the weighting parameters ω_1 , ω_2 , and ω_3 . Considering the absolute numerical value of each part of the reward, the alternative values of each parameter is set to [1, 3, 5], [1, 2, 3], and [4, 7, 10]. More precisely, a grid-based search can be programmed for the 27 situations. For each situation, an agent is trained in 300 episodes, and then 20 rounds of simulation are conducted to evaluate the performance of the agent.

Table 2 shows the metrics performance for different combinations of the parameters. It can be seen that an increase of ω_1 and ω_2 can improve the corresponding indicators to a certain extent in general, while the mean jerk is not sensitive to the change of ω_3 . Nevertheless, the increase of ω_2 does not necessarily reduce the travel time on some occasions. We infer that the weighting parameter with a high value encourages the agent to choose a more aggressive and greedy strategy, and this kind of strategy may be limited by the uncertain environment. The same is true of ω_1 . Besides, the mutual restriction of the three parameters will also exert an unexpected influence on the vehicle. Therefore, a compromising scheme is selected in this study. Concurrently, this provides plasticity for the control law. We can choose the parameter configuration with low travel time to improve the traffic efficiency if the travel is sensitive to the travel time. Similarly, we can also choose the configuration with low energy consumption to save electricity to the greatest extent. From Table 2, we can conclude that the energy consumption can be reduced by 31.73%~45.90%, while the sacrifice of travel time is 25%~98%.

Table 2. Results for different weighting parameters settings.

ω_1	ω_2	ω_3	Energy Consumption (Wh)	Travel Time (s)	Mean Jerk (m/s ³)
1	1	4	77.47	650.10	0.54
1	1	7	72.32	644.10	0.56
1	1	10	80.59	567.30	0.47
1	2	4	72.56	623.05	0.53
1	2	7	76.70	593.80	0.73
1	2	10	81.18	583.15	0.68
1	3	4	67.43	554.30	0.66
1	3	7	66.90	536.65	0.50
1	3	10	76.64	440.50	0.60
3	1	4	75.83	607.30	0.62
3	1	7	75.43	650.10	0.54
3	1	10	77.36	695.20	0.66
3	2	4	71.13	631.85	0.58
3	2	7	74.73	673.65	0.76
3	2	10	72.75	625.95	0.64
3	3	4	74.57	530.10	0.56
3	3	7	68.50	545.30	0.57
3	3	10	72.76	573.85	0.61
5	1	4	69.43	651.95	0.47
5	1	7	69.67	667.70	0.61
5	1	10	63.87	609.95	0.63
5	2	4	70.96	533.25	0.53
5	2	7	69.58	536.65	0.50
5	2	10	68.56	554.75	0.37
5	3	4	66.47	595.10	0.40
5	3	7	66.05	563.95	0.36
5	3	10	72.76	476.65	0.61

6. Conclusions

In this paper, we propose a reinforcement learning framework to implement eco-driving control for electric vehicles in the ATSC environment. The eco-driving problem is converted to an MDP because it is difficult to solve the optimal control problem directly.

The MDP has been carefully designed: the state of the vehicle includes not only its dynamic parameters but also the shared state of the signal agents, while the state sharing mechanism can help the ego vehicle make decisions in an uncertain environment. The reward function considers mobility, energy consumption, and comfort to enhance the driving strategy. Besides, an explanation module is established to explain the decision process of the ego vehicle, which can make the black-box model-free method clear and intuitive. The simulations are carried out on a signalized corridor built in SUMO. Being compared with RAINBOW and IDM, the proposed PPO algorithm gives a much better energy-saving performance. Despite the traffic mobility being reduced, this method can be used in circumstances where the requirement of travel time is not high, or energy is extremely scarce. Moreover, the SHAP value explicates the decision basis of the ego vehicle well, and this can provide reliability and safety for the actual circumstances.

It should be pointed out that the proposed strategy is still feasible for gasoline vehicles. Although we implement a RAINBOW-based ATSC, the whole framework can be applied to the ATSC environments controlled by some other algorithms.

This paper only considered an ideal V2I communication situation, but packet loss and communication delay are common in real world, so it is practical to study these conditions in future searches. Meanwhile, we only put forward the strategy for a single vehicle, while the corresponding methodology can be extended to platoon control, and this may be implemented in future research. Another research gap is that the interaction between vehicles with eco-driving systems and normal human-driven vehicles is not clear. Since the operation of eco-driving vehicles may have a direct impact on the holistic traffic flow, it is of value to explore the topic in a dedicated study. In addition, some data can be collected from a real-world vehicle operation system and used for training agents in an offline manner with offline DRL technique, which is helpful to formulate more applicable strategies to surmount the simulation-to-real transfer gap.

Author Contributions: Conceptualization, X.J. and J.Z.; methodology, X.J.; software, X.J. and B.W.; validation, X.J., J.Z. and B.W.; formal analysis, J.Z.; investigation, X.J. and B.W.; data curation, X.J. and B.W.; writing—original draft preparation, X.J. and J.Z.; writing—review and editing, X.J.; visualization, X.J.; supervision, J.Z.; project administration, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Key R&D Program of China (Grant 2021TFB-1600500), and the Key R&D Program of Jiangsu Province in China (Grant No. BE2020013). The work of the first author is supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. SJCX21_0062).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data in this study were produced through simulation experiments. Data sharing is not applicable to this article.

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

Appendix A

This part introduces the details for the implementation of the RAINBOW-based ATSC. The MDP is given in the body part of this paper, so we here present the information of the hyperparameter setting to enhance the reproducibility of the paper. The values of the algorithm-related parameters are shown in Table A1, while the meaning of the parameters can be found in the original paper [42].

For every episode of the training process, the simulator takes 300 s to load the vehicle into the road network and then train the agents. There are five traffic signal agents in accordance with Figure 6. Each agent learns independently, which means that it only considers its observation and action. The simulation duration of each episode is 3600 s,

and the training process takes a total of 50,000 s. Finally, the coverage curve is presented in Figure 3.

Table A1. Algorithm-related parameters of RAINBOW.

Description	Value
Multi-step returns	3
Distributional atoms	51
Distributional min/max values	[−50, 50]
Initial value of noisy net	0.5
Prioritization exponent	0.5
Prioritization importance sampling	0.4 → 1.0
Learning rate of the Adam optimizer	0.001

References

- Jiang, X.; Zhang, J.; Li, Q.-Y.; Chen, T.-Y. A Multiobjective Cooperative Driving Framework Based on Evolutionary Algorithm and Multitask Learning. *J. Adv. Transp.* **2022**, *2022*, 6653598. [\[CrossRef\]](#)
- Lakshmanan, V.K.; Sciarretta, A.; Ganaoui-Mourlan, O.E. Cooperative Eco-Driving of Electric Vehicle Platoons for Energy Efficiency and String Stability. *IFAC-PapersOnLine* **2021**, *54*, 133–139. [\[CrossRef\]](#)
- Siniša, H.; Ivan, F.; Tino, B. Evaluation of Eco-Driving Using Smart Mobile Devices. *Promet-Traffic Transp.* **2015**, *27*, 335–344. [\[CrossRef\]](#)
- Bakibillah, A.S.M.; Kamal, M.A.S.; Tan, C.P.; Hayakawa, T.; Imura, J.I. Event-Driven Stochastic Eco-Driving Strategy at Signalized Intersections from Self-Driving Data. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8557–8569. [\[CrossRef\]](#)
- Kamal, M.A.S.; Mukai, M.; Murata, J.; Kawabe, T. Ecological Vehicle Control on Roads with Up-Down Slopes. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 783–794. [\[CrossRef\]](#)
- Lee, H.; Kim, N.; Cha, S.W. Model-Based Reinforcement Learning for Eco-Driving Control of Electric Vehicles. *IEEE Access* **2020**, *8*, 202886–202896. [\[CrossRef\]](#)
- Zhang, R.; Yao, E.J. Eco-driving at signalised intersections for electric vehicles. *IET Intell. Transp. Syst.* **2015**, *9*, 488–497. [\[CrossRef\]](#)
- Li, M.; Wu, X.K.; He, X.Z.; Yu, G.Z.; Wang, Y.P. An eco-driving system for electric vehicles with signal control under V2X environment. *Transp. Res. Part C-Emerg. Technol.* **2018**, *93*, 335–350. [\[CrossRef\]](#)
- Wu, X.; He, X.; Yu, G.; Harmandayan, A.; Wang, Y. Energy-Optimal Speed Control for Electric Vehicles on Signalized Arterials. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2786–2796. [\[CrossRef\]](#)
- Zheng, Y.; Ran, B.; Qu, X.; Zhang, J.; Lin, Y. Cooperative Lane Changing Strategies to Improve Traffic Operation and Safety Nearby Freeway Off-Ramps in a Connected and Automated Vehicles Environment. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4605–4614. [\[CrossRef\]](#)
- Mintsis, E.; Vlahogianni, E.I.; Mitsakis, E. Dynamic Eco-Driving near Signalized Intersections: Systematic Review and Future Research Directions. *J. Transp. Eng. Part A-Syst.* **2020**, *146*, 15. [\[CrossRef\]](#)
- Nie, Z.; Farzaneh, H. Real-time dynamic predictive cruise control for enhancing eco-driving of electric vehicles, considering traffic constraints and signal phase and timing (SPaT) information, using artificial-neural-network-based energy consumption model. *Energy* **2022**, *241*, 122888. [\[CrossRef\]](#)
- Dong, H.; Zhuang, W.; Chen, B.; Lu, Y.; Liu, S.; Xu, L.; Pi, D.; Yin, G. Predictive energy-efficient driving strategy design of connected electric vehicle among multiple signalized intersections. *Transp. Res. Part C Emerg. Technol.* **2022**, *137*, 103595. [\[CrossRef\]](#)
- Liu, B.; Sun, C.; Wang, B.; Liang, W.; Ren, Q.; Li, J.; Sun, F. Bi-level convex optimization of eco-driving for connected Fuel Cell Hybrid Electric Vehicles through signalized intersections. *Energy* **2022**, *252*, 123956. [\[CrossRef\]](#)
- Asadi, B.; Vahidi, A. Predictive Cruise Control: Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time. *IEEE Trans. Control Syst. Technol.* **2011**, *19*, 707–714. [\[CrossRef\]](#)
- Lin, Q.; Li, S.E.; Xu, S.; Du, X.; Yang, D.; Li, K. Eco-Driving Operation of Connected Vehicle with V2I Communication Among Multiple Signalized Intersections. *IEEE Intell. Transp. Syst. Mag.* **2021**, *13*, 107–119. [\[CrossRef\]](#)
- Wang, Z.; Wu, G.; Barth, M.J. Cooperative Eco-Driving at Signalized Intersections in a Partially Connected and Automated Vehicle Environment. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 2029–2038. [\[CrossRef\]](#)
- Mousa, S.R.; Ishak, S.; Mousa, R.M.; Codjoe, J. Developing an Eco-Driving Application for Semi-Actuated Signalized Intersections and Modeling the Market Penetration Rates of Eco-Driving. *Transp. Res. Record* **2019**, *2673*, 466–477. [\[CrossRef\]](#)
- Yang, H.; Almutairi, F.; Rakha, H. Eco-Driving at Signalized Intersections: A Multiple Signal Optimization Approach. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 2943–2955. [\[CrossRef\]](#)
- Dong, H.; Zhuang, W.; Chen, B.; Yin, G.; Wang, Y. Enhanced Eco-Approach Control of Connected Electric Vehicles at Signalized Intersection with Queue Discharge Prediction. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5457–5469. [\[CrossRef\]](#)

21. Ma, F.W.; Yang, Y.; Wang, J.W.; Li, X.C.; Wu, G.P.; Zhao, Y.; Wu, L.; Aksun-Guvenc, B.; Guvenc, L. Eco-driving-based cooperative adaptive cruise control of connected vehicles platoon at signalized intersections. *Transport. Res. Part D-Transport. Environ.* **2021**, *92*, 17. [[CrossRef](#)]
22. Zhao, W.M.; Ngoduy, D.; Shepherd, S.; Liu, R.H.; Papageorgiou, M. A platoon based cooperative eco-driving model for mixed automated and human-driven vehicles at a signalised intersection. *Transp. Res. Part C-Emerg. Technol.* **2018**, *95*, 802–821. [[CrossRef](#)]
23. Zhao, X.M.; Wu, X.; Xin, Q.; Sun, K.; Yu, S.W. Dynamic Eco-Driving on Signalized Arterial Corridors during the Green Phase for the Connected Vehicles. *J. Adv. Transp.* **2020**, *2020*, 11. [[CrossRef](#)]
24. Rakha, H.; Kamalanathsharma, R.K. Eco-driving at signalized intersections using V2I communication. In Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, USA, 5–7 October 2011; pp. 341–346.
25. Mahler, G.; Vahidi, A. Reducing idling at red lights based on probabilistic prediction of traffic signal timings. In Proceedings of the 2012 American Control Conference (ACC), Montreal, QC, Canada, 27–29 June 2012; pp. 6557–6562.
26. Sun, C.; Shen, X.; Moura, S. Robust Optimal ECO-driving Control with Uncertain Traffic Signal Timing. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018; pp. 5548–5553.
27. El-Tantawy, S.; Abdulhai, B.; Abdelgawad, H. Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1140–1150. [[CrossRef](#)]
28. Li, L.; Lv, Y.S.; Wang, F.Y. Traffic Signal Timing via Deep Reinforcement Learning. *IEEE-CAA J. Autom. Sin.* **2016**, *3*, 247–254. [[CrossRef](#)]
29. Rasheed, F.; Yau, K.L.A.; Low, Y.C. Deep reinforcement learning for traffic signal control under disturbances: A case study on Sunway city, Malaysia. *Futur. Gener. Comp. Syst.* **2020**, *109*, 431–445. [[CrossRef](#)]
30. Liu, Y.; He, J. A survey of the application of reinforcement learning in urban traffic signal control methods. *Sci. Technol. Rev.* **2019**, *37*, 84–90.
31. Shi, J.Q.; Qiao, F.X.; Li, Q.; Yu, L.; Hu, Y.J. Application and Evaluation of the Reinforcement Learning Approach to Eco-Driving at Intersections under Infrastructure-to-Vehicle Communications. *Transp. Res. Record* **2018**, *2672*, 89–98. [[CrossRef](#)]
32. Mousa, S.R.; Ishak, S.; Mousa, R.M.; Codjoe, J.; Elhenawy, M. Deep reinforcement learning agent with varying actions strategy for solving the eco-approach and departure problem at signalized intersections. *Transp. Res. Record* **2020**, *2674*, 119–131. [[CrossRef](#)]
33. Guo, Q.Q.; Angah, O.; Liu, Z.J.; Ban, X.G. Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors. *Transp. Res. Part C-Emerg. Technol.* **2021**, *124*, 20. [[CrossRef](#)]
34. Wegener, M.; Koch, L.; Eisenbarth, M.; Andert, J. Automated eco-driving in urban scenarios using deep reinforcement learning. *Transp. Res. Pt. C-Emerg. Technol.* **2021**, *126*, 15. [[CrossRef](#)]
35. Zhang, X.; Jiang, X.; Li, N.; Yang, Z.; Xiong, Z.; Zhang, J. Eco-driving for Intelligent Electric Vehicles at Signalized Intersection: A Proximal Policy Optimization Approach. In Proceedings of the ISCTT 2021, 6th International Conference on Information Science, Computer Technology and Transportation, Xishuangbanna, China, 26–28 November 2021; pp. 1–7.
36. Zhang, J.; Jiang, X.; Cui, S.; Yang, C.; Ran, B. Navigating Electric Vehicles Along a Signalized Corridor via Reinforcement Learning: Toward Adaptive Eco-Driving Control. *Transp. Res. Record* **2022**, 03611981221084683. [[CrossRef](#)]
37. Ouyang, Q.; Wang, Z.; Liu, K.; Xu, G.; Li, Y. Optimal Charging Control for Lithium-Ion Battery Packs: A Distributed Average Tracking Approach. *IEEE Trans. Ind. Inform.* **2020**, *16*, 3430–3438. [[CrossRef](#)]
38. Liu, K.; Li, K.; Zhang, C. Constrained generalized predictive control of battery charging process based on a coupled thermoelectric model. *J. Power Sources* **2017**, *347*, 145–158. [[CrossRef](#)]
39. Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. In Proceedings of the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4768–4777.
40. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wiessner, E. Microscopic Traffic Simulation using SUMO. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2575–2582.
41. Wang, Y.Z.; Yang, X.G.; Liang, H.L.; Liu, Y.D. A Review of the Self-Adaptive Traffic Signal Control System Based on Future Traffic Environment. *J. Adv. Transp.* **2018**, *12*, 1096123. [[CrossRef](#)]
42. Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
43. Kurczveil, T.; López, P.Á.; Schnieder, E. Implementation of an Energy Model and a Charging Infrastructure in SUMO. In Proceedings of the Simulation of Urban MObility User Conference, Berlin, Germany, 15–17 May 2013; pp. 33–43.
44. Kesting, A.; Treiber, M.; Helbing, D. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philos. Trans. Royal Soc. A* **2010**, *368*, 4585–4605. [[CrossRef](#)]
45. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
46. Ying, C.S.; Chow, A.H.F.; Wang, Y.H.; Chin, K.S. Adaptive Metro Service Schedule and Train Composition with a Proximal Policy Optimization Approach Based on Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–12. [[CrossRef](#)]

47. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
48. Liu, K.; Peng, Q.; Li, K.; Chen, T. Data-Based Interpretable Modeling for Property Forecasting and Sensitivity Analysis of Li-ion Battery Electrode. *Automot. Innov.* **2022**, *5*, 121–133. [[CrossRef](#)]
49. He, L.; Aouf, N.; Song, B. Explainable Deep Reinforcement Learning for UAV autonomous path planning. *Aerosp. Sci. Technol.* **2021**, 107052. [[CrossRef](#)]
50. Ribeiro, M.T.; Singh, S.; Guestrin, C. “Why should i trust you?” Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
51. Lundberg, S.M.; Nair, B.; Vavilala, M.S.; Horibe, M.; Eisses, M.J.; Adams, T.; Liston, D.E.; Low, D.K.-W.; Newman, S.-F.; Kim, J.; et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nat. Biomed. Eng.* **2018**, *2*, 749–760. [[CrossRef](#)] [[PubMed](#)]